GRAVITY GRADIOMETRY AND MAP MATCHING:

AN AID TO AIRCRAFT INERTIAL NAVIGATION SYSTEMS

THESIS

Anthony DeGregoria, Major, USAF

AFIT/GAE/ENY/10-M06

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/10-M06

GRAVITY GRADIOMETRY AND MAP MATCHING: AN AID TO AIRCRAFT INERTIAL NAVIGATION SYSTEMS

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

Anthony DeGregoria, BS

Major, USAF

March 2010

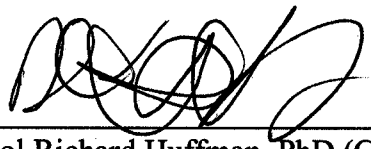
AFIT/GAE/ENY/10-M06

GRAVITY GRADIOMETRY AND MAP MATCHING: AN AID TO AIRCRAFT INERTIAL NAVIGATION SYSTEMS

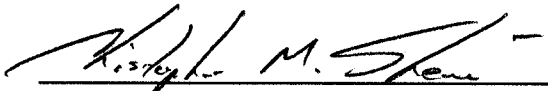
Anthony DeGregoria, BS

Major, USAF

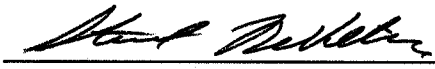
Approved:



Lt Col Richard Huffman, PhD (Chairman) — 18 Mar 2010 — Date



Lt Col Christopher Shearer, PhD (Member) — 18 Mar 2010 — Date



Dr. Stewart DeVilbiss, PhD (Member) — 18 Mar 10 — Date

Abstract

Inertial navigation systems (INS) offer passive, all-weather, and undeniable navigation information, which military customers often view as especially appealing strengths. Unfortunately, Airmen and engineers still struggle with INS's drifting position errors, and navigation aids generally detract from INS's strengths. At this year's Air, Space, and Cyberspace in the 21st Century Conference, the Chief of Staff of the Air Force identified the Global Positioning System (GPS) as a widely-known and exploitable vulnerability, saying that it's critical the Joint force reduce GPS dependence. Recent advances provide an opportunity for gravity gradient instruments (GGI), which measure spatial derivatives of the gravity vector, to aid an INS and preserve its strengths.

This thesis shows that a GGI and map matching enhanced (GAME) INS improves navigation accuracy, presents the conditions that make GAME feasible for aircraft, and identifies opportunities for improvement. The methodology includes computer models and algorithms, where a GGI and map matching aid an INS through a Kalman filter. Simulations cover different terrains, altitudes, velocities, flight durations, INS drifts, update rates, components of the gravity gradient tensor, GGI and map noise levels, map resolutions, and levels of interpolation. Although GAME with today's technology only appears worthwhile for long range and long endurance flights, the technologies expected in 10 years promise a broad spectrum of scenarios where GAME potentially provides great returns on investments and dominates the market for secure and covert navigation.

Acknowledgments

This paper is dedicated to those who have lost their way and don't ask for help. Whether you're afraid, alone, or ashamed; too angry, too proud, or too depressed; unable, unselfish, or you feel undeserving. We are all children of the same God, and we all want you to come home. The Creators of heaven and earth always provide an inner voice that leads you back to the true path.

Thank you to my Father in heaven, my family at home, and my friends at AFIT. This work would not be possible without you. You provided me with information, inspiration, and infinite support. Thank you to Dr. Stewart DeVilbiss and Dr. Jacob Campbell for sponsoring this work. Thank you to Lieutenant Colonel Richard Huffman, PhD., for leading this effort and tuning me in to gravity gradients, Lieutenant Colonel Michael Veth, PhD., for pointing me in the right direction, Lieutenant Colonel Christopher Shearer, PhD., for keeping me honest, and Captain Marshall Rogers for providing the first maps on how to get there.

Finally, I would like to thank my wife, who listened to my techno-babble and fully supported long hours of study. I think it's time we take a vacation… together this time!

Anthony DeGregoria

Table of Contents

List of Figures

Page

List of Tables

List of Equations

Nomenclature

AFIT	Air Force Institute of Technology
BEP	Break Even Point
CEP	Circular Error Probable (50th percentile)
CSAF	Chief of Staff of the Air Force
DTED	Digital Terrain Elevation Data
EGM96	Earth Gravitational Model 1996
Eö	Eötvös (a unit of measure for gravity gradients; 1 Eö = 10^{-9} s^{-2})
FFT	Fast Fourier Transform
GAME	Gravity Gradiometry and Map Matching Enhanced
GGI	Gravity Gradient Instrument
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISR	Intelligence, Surveillance, and Reconnaissance
ND	Non-Dimensional
NED	North, East, Down (a coordinate reference frame)
NSD	Noise Spectral Density
PDF	Probability Density Function
PMF	Point Mass Filter
RMS	Root Mean Squared
TERCOM	Terrain Contour Matching
TRN	Terrain Referenced Navigation

GRAVITY GRADIOMETRY AND MAP MATCHING: AN AID TO AIRCRAFT INERTIAL NAVIGATION SYSTEMS

I. Introduction

Since the first integration of an inertial navigation system (INS) on an aircraft, aviators and engineers pursued improvements in navigation accuracy. The demand for navigation accuracy outpaced advances in INS technologies and quickly motivated the search for new ways to aid the INS. Some of the ideas included integrating information provided by the aircrew's visual observation of landmarks, Global Navigation Satellite Systems (GNSS) such as the Global Positioning System (GPS), and terrain referenced navigation (TRN). Most external aids focused on providing accurate position information, since this was a critical weakness of the INS. Unfortunately, the use of external aids generally detracted from some of the INS's most appealing strengths, especially from the perspective of military customers: the INS's autonomy, all-weather capability, and low vulnerability.

For example, the visual observation of landmarks requires adequate visibility and time-consuming work by the aircrew. By definition, TRN matches terrain maps to radar altimeter measurements, thus requiring the emission of signals that enemy personnel or radar guided missiles might detect. Currently, GPS stands as the preferred complement to INS, but relies on an external constellation of satellites, which are vulnerable to destruction and whose signals are vulnerable to interference and jamming. Even today, after over 60 years on the market, aviators and engineers aggressively pursue improvements to the accuracy of the INS, especially including the development and integration of external aids that preserve the aforementioned strengths of the INS.

The turn of another century, however, brings significant advances in accelerometer technologies, including Gravity Gradient Instruments (GGIs). New ideas for employing accelerometers, improving sensitivity, and reducing noise now make GGIs a capable navigation aid. Although the integration of GGIs into navigation systems is still in its infancy, engineers in several places have already taken the first steps. The mining industry flies GGIs onboard aircraft to rapidly survey the geology, the Navy pursues covert submarine navigation, and academics publish papers and patents; all with gravity gradiometry as an aid for navigation. Now it's time to investigate the feasibility of using gravity gradiometry and map matching as an aid to aircraft inertial navigation systems.

Problem Statement

America's Air and Space Forces need a navigation aid that provides accurate position information and preserves the strengths of the INS; namely the autonomy, all-weather capability, and low vulnerability that military customers desire. The popularity of the INS in aircraft testifies to its value as a navigation aid. However, the pursuit of improving INS position accuracy is almost as popular. Current systems that aid the INS and improve position accuracy generally detract from INS's previously mentioned strengths. Some might argue that one of the most popular complements to the INS, GPS, meets these needs, but the Chief of Staff of the Air Force clearly identifies a concern that enemies may possess the potential to deny GPS information, and it's critical the Joint force reduce GPS dependence.[1] Furthermore, no backup external aid appears to exist, which can provide GPS-level navigation accuracy.

The *Literature Review* suggests that GGI technologies might be capable of meeting these needs in the foreseeable future, but relatively little research exists about using gravity gradiometry as a navigation aid. Some pioneers in this field show that GGIs can improve position accuracy by providing information that reduces the INS's errors due to estimates of the gravity vector. Although this approach yields significant improvement, it does not eliminate the tendency of the INS to drift over time. Others show that information from GGIs can be matched to a gravity gradient map to determine a position. However, nothing in open literature provides a comprehensive assessment of the capabilities of a navigation system that uses gravity gradiometry and map matching to aid an aircraft INS. Comprehensive, in this case, refers to an assessment that shows the potential for this concept to provide certain levels of navigational performance, while considering variations in key parameters. These key parameters might include terrain, altitude, velocity, flight duration, INS drift rate, position update rate, GGI performance, and map resolutions and accuracies. A valuable assessment of capabilities would define the conditions for which gravity gradiometry and map matching become a feasible aid to an aircraft INS.

This paper will focus on gravity gradiometry and map matching as an aid to the aircraft INS. Can this concept improve navigation performance? What conditions or values of key parameters make this concept feasible? What research and advances in technology might improve the performance of this concept? Answering these questions might uncover a navigation aid that provides accurate position information while preserving the strengths of the aircraft INS.

Research Objectives

As a student of the Air Force Institute of Technology, my research supports the organization's vision to "sustain the technological supremacy of America's Air and Space Forces". In the spirit of this vision, these three overarching objectives guide my research:

1. Show that a gravity gradiometry and map matching enhanced (GAME) aircraft inertial navigation system can improve navigation accuracy.
2. Determine what conditions make the GAME feasible for aircraft navigation.
3. Identify what research and advances in technology improve the GAME.

In order to show that GAME can improve navigation accuracy, this research gives a clear description of the work previously done to aid aircraft inertial navigations systems with GGIs. Next, this research develops an algorithm that matches real-time GGI signals to a location on a gravity gradient map. Simulations demonstrate that the algorithm can process GGI signals, match the signals to a gravity gradient map, determine a position, and provide useful information to an aircraft INS in real time. To fulfill the first objective, this research reports GAME's accuracy by comparing position solutions to the aircraft's true positions.

The effort to determine what conditions make GAME feasible for aircraft navigation starts with a sensitivity analysis, which shows the effects of key parameters on GAME performance. Additional simulations ensure that results address practical scenarios, which allow an easier assessment of GAME's potential impact on the Air Force mission. Depending on the lessons learned from the literature review and the

constraints of this research effort, key parameters might include variations in terrain, altitude, velocity, flight duration, INS drift rate, position update rate, GGI performance, and map resolutions and accuracies. Standardized navigation performance metrics ensure the effects of variations in key parameters can be quantified and easily compared. A summary of results shows how key parameters affect GAME's accuracy, thereby providing a tool for determining what conditions make GAME feasible for aircraft navigation.

Finally, with knowledge gained from completion of the previous two objectives, this research identifies what further research and advances in technology will drive the greatest improvements in GAME accuracy. Ideally, this list will be prioritized and potentially guide future efforts along a better, faster, and cheaper path, making GAME a valuable aid to aircraft inertial navigation systems.

II. Literature Review

Inertial Navigation Systems

Traditionally, an inertial navigation system (INS) produces navigation information by processing signals from accelerometers and gyros with a computer. In its most basic terms, an INS performs inertial navigation, or the determination of the position and velocity of a moving object, by using instruments that sense motion. People commonly refer to these instruments as inertial measurement units (IMUs), which includes accelerometers to sense linear acceleration and gyroscopes to sense angular rates. A simple INS, as shown in Figure 1, orients three accelerometers and three gyroscopes orthogonally and straps them down to a stable platform with a computer. The placement and orientation of the IMUs and their platform provide the basis of a reference frame. Thus, the INS computer can use a transformation matrix to turn integrated accelerometer measurements and orientation information from the gyroscopes into useful information in the navigational reference frame. Of course, this description represents only a simple portrayal of an INS, and many variations to these concepts exist.[2]

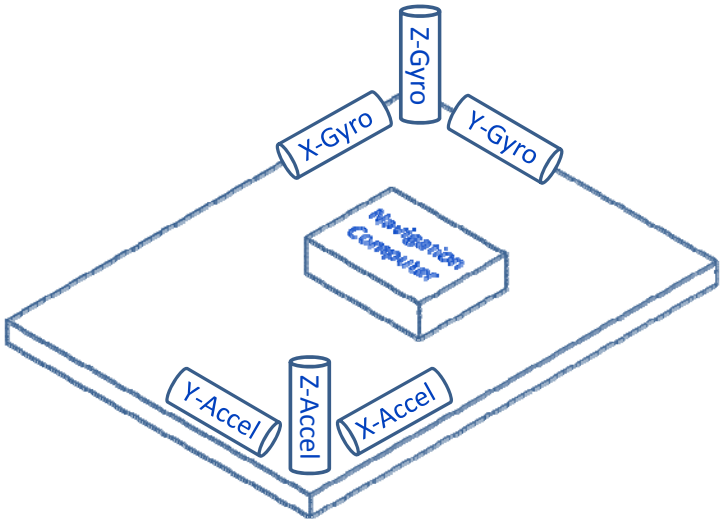


Figure 1. A Simple Inertial Navigation System

The unique strengths of an INS include its measurements, autonomy, and low-vulnerability, making it one of the most popular aircraft navigation systems today. The IMUs measure the derivatives of position, velocity, and attitude at high sampling rates, which ideally suits INS for integrated navigation, guidance, and control. Additionally, since INS independently measures inertia, it provides useful navigation over random routes without the use of any external aids or visual cues. The history of INS proves its reliability and shows that it functions worldwide, including underwater, on land, in tunnels, buildings, or containers, in the skies all around the Earth, and in space. By its nature, enemies cannot detect or jam INS, because it does not transmit detectable signals and requires no external windows or antennas. Furthermore, the independence of INS means that enemies cannot deny a user of information from an INS, because there are no third party transmitters, receivers, ground facilities, or satellites to attack. Some might describe INS as the ultimate in military stealth navigation.[3,4]

Inertial Navigation System Errors

Navigational errors have been a problem since the first nomad got separated from the masses. Similarly, people have been pursuing reductions in INS errors since they first used accelerometers and gyroscopes for navigation. The gyroscope became a suitable substitute for the magnetic compass to dead reckon ships around 1911, eventually leading to the automatic steering of ships in the 1920's. However, assembling the modern-day INS for use on an aircraft took a little more time and started off with significant errors, as detailed in the following account from Ernst Steinhoff:

> "In 1930 an attempt was made to navigate an aircraft equipped with a gyrostabilized platform and mechanically integrating accelerometers mounted on it. The flight, which departed from Berline-Aldersdorf, was discontinued and the attempt terminated after three hours of flying time when the aircraft equipment indicated a position somewhere in Australia, while visual observations confirmed the aircraft position to be at the western border of Germany near Holland."[5] (p.47)

Thanks to rocket scientists like Robert H. Goddard in the 1930's and those who worked in the World War II and Cold War eras, INS errors quickly decreased. German rockets in World War II eventually found their way to England and the Cold War drove a steady reduction in errors. In 1959, the Atlas D intercontinental ballistic missile possessed a circular error probable (CEP) of about 1.8 nautical miles (NMI), while the Minuteman 3 achieved a 0.21 NMI CEP in 1970, and the MX a 0.06 NMI CEP in 1986.[4] IMU and INS errors continued to decrease and, today, are generally considered to possess the accuracies shown in the following table:

Table 1: INS, Gyro, and Accelerometer Performance Ranges (Data from Ref 2)

	Units	*High Performance*	*Medium Performance*	*Low Performance*
INS	nmi/hr	$\leq 10^{-1}$	≈ 1	≥ 10
Gyro	deg/hr	$\leq 10^{-3}$	$\approx 10^{-2}$	$\geq 10^{-1}$
Accelerometer	m/s^2	$\leq 10^{-6}$	$\approx 10^{-5}$	$\geq 10^{-4}$

An understanding of the source of INS errors begins with an understanding of the navigation equations for position. The foundation of inertial navigation rests on knowing the magnitude and direction of an object's accelerations, denoted by the vector $\ddot{\vec{x}}$, which

then leads to the position of the object, $\vec{x}$, after integrating with respect to time twice. The acceleration of the object, $\ddot{\vec{x}}$, equals the gravitational field, $\vec{g}(\vec{x})$, plus the measured acceleration, $\vec{a}$.

$$\ddot{\vec{x}} = \vec{g}(\vec{x}) + \vec{a}$$

Equation 1: Navigation Equation for Position

Perturbation of Equation 1 produces a first order approximation of the total error in the acceleration of the object, $\delta\ddot{\vec{x}}$, where $\delta\vec{g}$ represents the error in the assumed gravitational field, $\delta\vec{x}$ represents the position error of the object, and $\delta\vec{a}$ represents the error in the measured acceleration.

$$\delta\ddot{\vec{x}} = \frac{\delta\vec{g}}{\delta\vec{x}}\delta\vec{x} + \delta\vec{g} + \delta\vec{a}$$

Equation 2: First Order Navigation Equation for Position Error

The term $\frac{\delta\vec{g}}{\delta\vec{x}}$ represents the gravity gradients, a second order tensor of the gravitational field's partial derivatives with respect to the coordinate system that defines the position.[2] The *Gravity Gradients* section discusses more about this second-order tensor.

For analytical purposes, INS errors break down into three general categories: initial alignment errors, inertial sensor errors, and computational errors. Since inertial

navigation systems integrate information from the past to identify the current position, all errors introduced to the system remain in the system and accumulate over time. [7] Although some errors remain constant or oscillate over time, gyroscope bias and initial heading errors generally cause the overall position error of the INS to increase over time. The term *drift* refers to the sum of all position errors, since the INS's calculated position appears to drift relative to the true position as time passes. According to the 2006 Aviator's Guide to Navigation, a drift rate of 2 nautical miles per hour is the traditional aircraft industry standard, although the advent of ring laser gyroscopes reduced INS drift rates to about 0.2 nautical miles per hour.[4] Figure 2 illustrates the typical magnitude and behavior of the sources of position error for an aircraft INS over time.

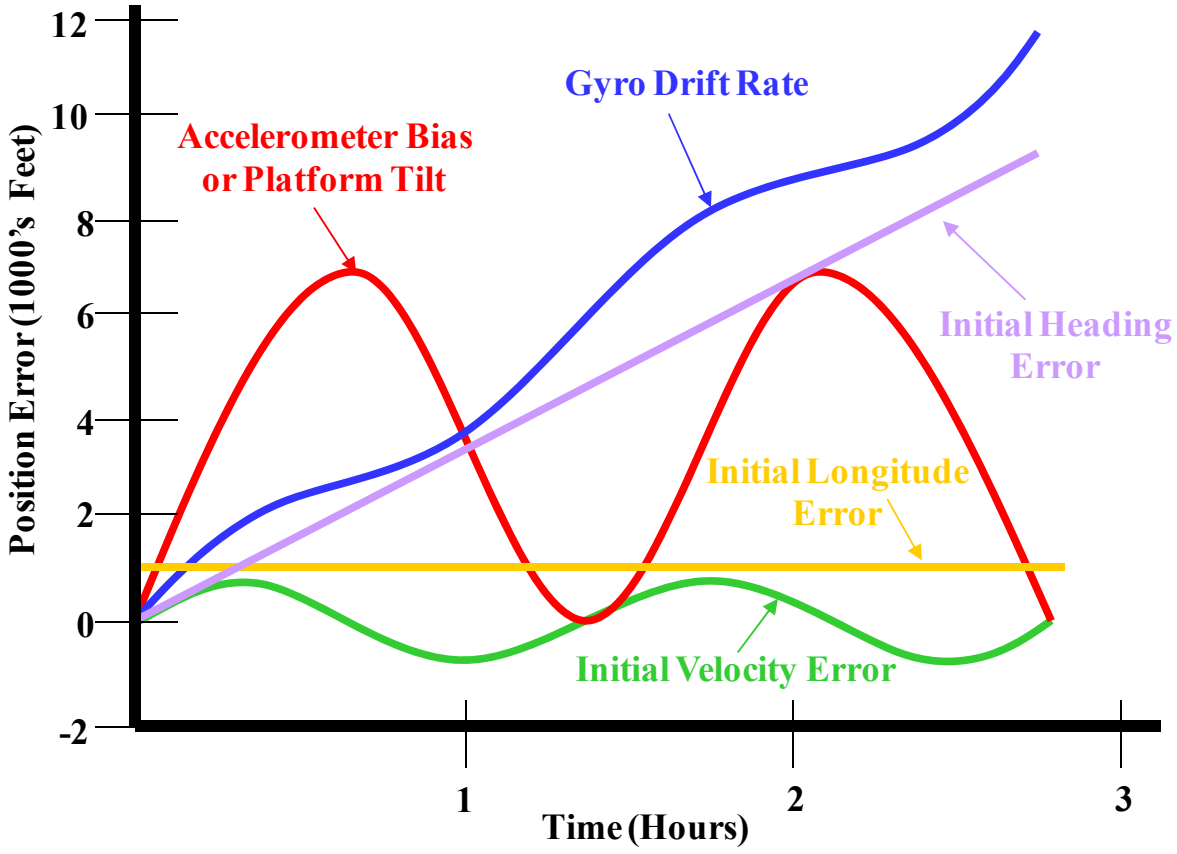


Figure 2: Breakdown of Position Error Sources for a Typical Aircraft INS[6]

Figure 2 assumes an INS initial alignment or position update occurred at a time of zero hours, which initializes the position errors related to each error source at nearly zero with the exception of the initial longitude error. For short durations of flight (e.g. less than 0.25 hours), no source of error appears to dominate. For medium duration flights (e.g. 0.5-3 hours), a combination of the accelerometer bias, initial heading error, and gyro drift rate (previously referred to as gyroscope bias) appears to dominate. For long duration flights (e.g. greater than 3 hours), the gyro drift rate grows dominant. For this reason, some might identify the performance of the gyroscope as the critical factor in achieving long-term system accuracy. In fact, Titterton uses the performance of the gyroscope as a rule of thumb to estimate INS drift rates (i.e. gyros with 0.01 degrees per hour of error should result in an INS with approximately 1 kilometer per hour of drift).

Since INS's drift over time, it's important to track how the sources of error affect each other and how their magnitudes and directions change as they propagate forward in time. To accomplish this, Titterton derives first order equations to estimate the sources of error at a given time, given the initial sources of error. These initial sources of error include the tilt error, heading error, velocity error, position error, gyroscope bias, and accelerometer bias. At any given time, the current error may be found by multiplying a transition matrix by the assumed initial error source: $[\delta x(t)] = [\Phi(t - t_o)]\,[\delta x(t_o)]$. The matrix $[\delta x(t)]$ possesses the same elements as $[\delta x(t_o)]$.[7]

Two other important variables that affect INS position solutions include the Schuler Frequency, ω_S, and the rate of rotation, $\dot{\Lambda}$. The Schuler Frequency represents the oscillation of horizontal errors attributed to the tuning of an INS such that it maintains proper orientation despite accelerations in the horizontal direction. On Earth, the Schuler

Frequency is given by $\omega_s = \sqrt{\frac{g}{R_o}}$, where g is the magnitude of the gravitational vector and R_o represents the Earth's radius of curvature. The period of these oscillations equal about 84.4 minutes. Titterton desrcibes Schuler tuning as part of "the first steps towards all-weather, autonomous navigation" (p.12). The rate of rotation affects the propagation of initial attitude error and gyroscope bias and is given by $\dot{\Lambda} = \Omega + \frac{V_E}{R_o \cos L}$, where Ω represents the Earth's rate of rotation, V_E represents the east velocity, and L represents the latitude. [7]

In summary, INS position errors arise from initial alignment errors, inertial sensor errors, and computational errors. Complex sets of coupled equations, however, can estimate INS drift over time. Sometimes the equations simplify, especially considering short duration flights and/or benign flight environments. But for precision aircraft navigation, many sources of error must be considered, including many that were not mentioned. The sources of error must be estimated and propagated through time, in order to predict the uncertainty in the INS's calculated position.[7]

Aiding Inertial Navigation Systems

The previous section showed that the accuracy of the position calculated by the INS drifts over time, due to initial alignment errors, inertial sensor errors, and computational errors. More accurate sensors and faster computer processors directly reduce these errors and improve INS accuracy, but, at some point, further improvements in these technologies become too expensive or unfeasible. This is when other navigation

methods might be called upon to aid the INS, because of their feasibility, reasonable costs, or other benefits such as system redundancy or even improvements in some performance characteristics beyond those achievable by the INS. In general, the navigation methods that aid the INS can be categorized as those that use external and onboard measurements. Navigation aids that use external measurements include information obtained from radio transmitters, satellites, stars, ground-based vehicles and stations, and visually observed landmarks. Navigation aids that use onboard measurements include altimeters, radars, airspeed indicators, magnetic sensors, and electro-optical imaging systems. Note that Titterton's 2004 comprehensive textbook doesn't mention gravity gradient instruments and map matching as a possible navigation aid.[7] This section discusses the traditional INS navigation aids, while the rest of this paper focuses on gravity gradiometry.

To improve INS accuracy, the information gained from one or more navigation aids must be integrated with the INS. This integration of information may be loosely coupled, tightly coupled, or remain uncoupled. *Loosely coupled* refers to a system where information from the aid feeds into and improves the INS's performance, but both navigation systems retain their own data processing algorithms. *Tightly coupled* refers to a system where information feeds from the INS and aids into a single processor, which then optimizes usage of the information to maximize navigation accuracy and improves the performance of the individual navigation systems. *Uncoupled* refers to a system where no information from the aid feeds back to the INS to improve its performance. In all of these cases, because information from two or more different navigation systems feeds into a navigation solution (e.g. the INS and its aid), the overarching system is

referred to as an *integrated navigation system*. Figure 3 provides a conceptual illustration of an integrated navigation system that tightly couples an INS with a navigation aid.

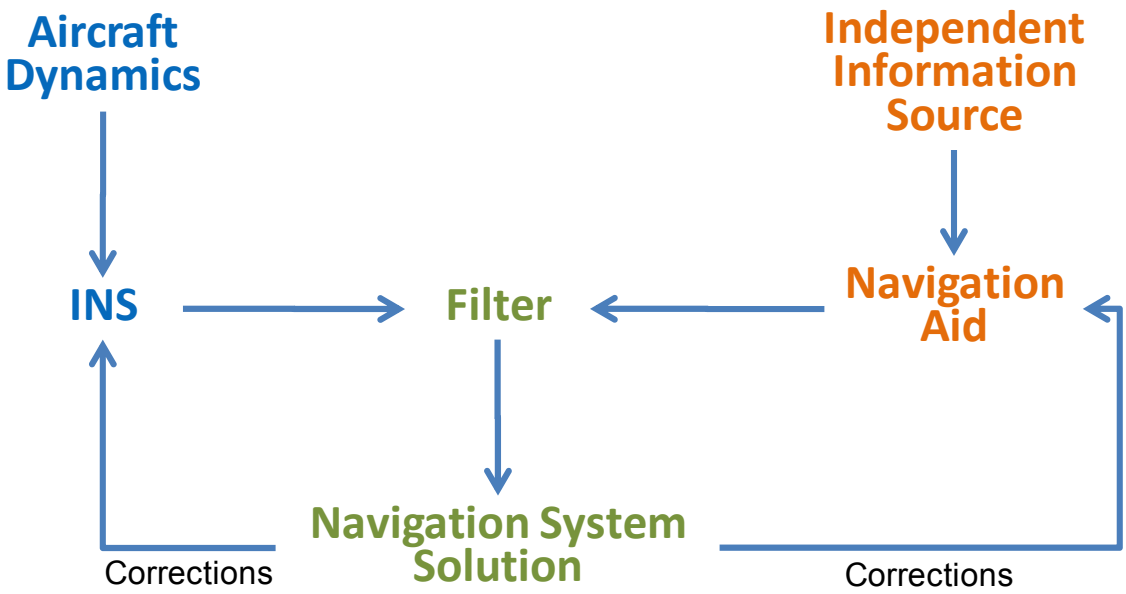


Figure 3: Tightly Coupled Integrated Navigation System

A closer look at one of today's most popular INS aids, the Global Positioning System (GPS), provides an excellent example of the potential benefits of an integrated navigation system. After processing, INS provides stable position, velocity, and attitude information at high data rates. Unfortunately, INS errors accumulate over time, resulting in good short-term performance, but long-wavelength errors and boundless drift. On the other hand, GPS provides position, velocity, and time information at slower data rates. GPS produces discrete information, so errors do not accumulate over time. However, data rates tend to be slower and depend on a network of satellites and a ground segment.[8] While INS and GPS methods appear to possess opposite strengths and weaknesses, this is

precisely why they form one of the most popular foundations for integrated navigation. As an integrated system, INS and GPS possess the strengths to make up for the other's weaknesses. For example, GPS provides added value to INS in terms of periodic position and velocity updates. Simultaneously, INS provides attitude information to help GPS locate satellites and short-term information at high frequencies to help GPS overcome signal losses and cycle slips in the phase ambiguity resolution process, which is especially helpful to GPS receivers in high-dynamic flight environments. Table 2 summarizes some of the difference between navigation with GPS and INS observed by Jekeli.

Table 2: Differences between Navigation with GPS and INS[2]

	GPS	*INS*
Principle	Time Delay of Signals	Inertia
Outputs	Position & Time	Position & Orientation
Error Wavelengths	Short	Long
Data Rate	Low	High
Dependence	Ground & Space Segments	Autonomous

When one or more aids work with an INS, a computer algorithm must integrate the information and ideally maximize the accuracy of the navigation solution. Many algorithms have been developed to reduce or bound INS errors, sometimes by simply updating the INS to a new position based on information from an aid. Today, however,

the Kalman filter stands as the algorithm of choice for integrating information from an INS and other aids. Titterron describes the Kalman filter as "the combination of two estimates of a variable to form a weighted mean, the weighting factors chosen to yield the most probable estimate" (p.385). In the case of an aided INS, one estimate derives from INS information and the equations of motion, while the second comes from an aid. The Kalman filter entered aerospace when Rudolf E. Kalman presented his linear, state-space dynamics modeling filter theory at the National Aeronautics and Space Administration in 1960. The first famous application of the Kalman filter occurred on the Apollo moon flight, where it provided midcourse navigation corrections, which least squares fitting techniques previously accomplished at the expense of the largest and best computers of the time. Quickly, engineers modified Kalman's filter to iteratively linearize about the current state. This allowed the Kalman filter to handle nonlinear dynamics and became known as the *extended Kalman filter*. The power of the Kalman filter in aided INS applications rests on the fact that it solves several inertial navigation problems efficiently. According to Biezad, these include

> "how to correct the navigation error equations while flying so that they remain useful even though the initial navigation errors were not known accurately; how to deal with noisy measurements from a variety of other systems that are arriving at different times; how to estimate the covariance of the INS output whenever an update occurs to see how much of the measurement should be believed in the presence of noisy system dynamics; and finally, how to obtain estimates for all navigation outputs even though only one or two is being measured by other means, providing as a result the most probable position (MPP)." (p. 97) [8]

Equation 3 includes the fundamental matrix equations of a discrete, linear Kalman filter. The first equation defines a system model, where *x* represents the true or actual state variables, Φ represents the model that propagates the state variables to the next time step, *w* represents the difference between the model and truth, and *k* represents a given time step. The second equation defines a measurement model, where *z* represents the measurement data, *H* represents the model that relates state to measurement variables, and *v* represents the residual error. The remaining equations define the state variable estimates, $\hat{x}$, the error covariance extrapolation, *P*, and the Kalman gain, *K*. These equations use (-) to indicate variables that do not consider the k^{th} data point and (+) to indicate variables that include the k^{th} data point. *Q* represents a matrix of covariances that define the system's noise, while *R* represents the measurement noise. These equations assumed no correlation between the system and measurement noise and that both possess a zero mean and Gaussian distribution.[9]

System Model:	$x_k = \Phi_{k-1} x_{k-1} + w_{k-1}$
Measurement Model:	$z_k = H_k x_k + v_k$
State Estimate Extrapolation:	$\hat{x}_k(-) = \Phi_{k-1} \hat{x}_{k-1}(+)$
Error Covariance Extrapolation:	$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^T + Q_{k-1}$
State Estimate Update:	$\hat{x}_k(+) = \hat{x}_k(-) + \bar{K}_k[z_k - H_k \hat{x}_k(-)]$
Error Covariance Update:	$P_k(+) = [I - \bar{K}_k H_k] P_k(-)$
Kalman Gain Matrix:	$\bar{K}_k = P_k(-) H_k^T [H_k P_k(-) H_k^T + R_k]^{-1}$

Equation 3: Discrete Linear Kalman Estimator

Although a presentation of other Kalman filter equations, derivations, and modifications are beyond the scope of this paper, Chatfield provides a detailed development of the linear and extended Kalman filter navigation and error equations.[10] Biezad, Jekeli, and Titterton present similar information. Users must remain aware of Kalman filter pitfalls. Kalman filters use linearized eauations based on small perturbation theory, so large errors or corrections to the system could result in divergence from real world behavior. Also, system integrity could be lost if the covariance matrix becomes too small. Biezad describes this as "Kalman filter incest," where the filter essentially rejects new measurements in favor of estimates propagated from the past. This danger can be summarized as the filter rejecting new/good measurements in favor of old data that becomes less accurate with time. On the other hand, if the filter is too liberal, then it will give greater weight to less accurate measurements (i.e. accept bad measurements). Both of these situations lead to a loss of system integrity. Although engineers continue to deal with these pitfalls, solutions exist that provide high reliability for the Kalman filter in aided INS applications.

Historically, many different technologies have come to the aid of INS, each with its strength and weaknesses. While visual identification of landmarks and stars represent some of the oldest navigation aids, radio aids might be described as the oldest, modern-day navigation aid. The transmission and reception of radio waves, often involving ground stations, allows airborne receivers to determine range and/or distance relative to the known location of transmitters. Some of these include very high frequency omni-directional radio range (VOR) and tactical air navigation (TACAN). Hyperbolic systems, like Decca, Omega, and long range navigation (LORAN), generally rely on signals

transmitted from two or more ground stations at the same time, which allows the receiver to estimate a position relative to the known ground stations based on the difference in the time of arrival of the signals at the receiver. Global Navigation Satellite Systems (GNSS), such as the Global Positioning System (GPS), rely on a network of satellites, which transmit radio signals that allow the receiver to triangulate its position. Depending on the system's sophistication and involvement of ground stations, radars have the capability of providing range, elevation, and bearing information between the aircraft and a known location.

Altimeters, including barometric, radar, and radio, provide height measurements, which help the INS bound errors in the vertical direction. Radar altimeter applications were broadened to include a primary role in terrain referenced navigation (TRN), where the altimeter provides a ground profile beneath the aircraft's flight path, which is then compared to terrain contour maps to determine a position. The roughness of the terrain and ground cover (e.g. snow and trees) affect the accuracy of this method. Another TRN method compares terrain height estimates to the contour map. The difference between INS and altimeter heights provides the estimate for comparison.[7] An extension of this idea might compare changes in terrain height over an estimated distance traveled along the map (Δh_{map}) to an estimated difference in the terrain height (Δh_{est}), which comes from differencing altimeter (Δh_{alt}) and INS (Δh_{ins}) measurements. The section on *Map Matching Algorithms* includes examples of more TRN concepts.

$$\Delta h_{est} = \Delta h_{alt} - \Delta h_{ins}$$

Equation 4: Terrain Height Differencing

Some navigation aids do not require the transmission of manmade signals external to the aircraft. While magnetic measurements, such as the compass, have provided bearing information for hundreds of years, possibilities now exist to determine attitude and position. These possibilities use instruments that measure multiple directional components of the Earth's magnetic field and stored maps of the Earth's magnetic variations. Scene matching area correlation (SMAC) determines a position, including altitude, by using a correlation algorithm to compare an image of the ground to a stored database of ground features.

This review of INS, including errors and aids, provides a foundation for understanding how gravity gradiometer instruments might play a role in navigation. Richeson illustrates the concept of a coupled gravity gradient instrument (GGI) and INS in Figure 4. The following sections focus on the fundamentals of gravity gradiometry.

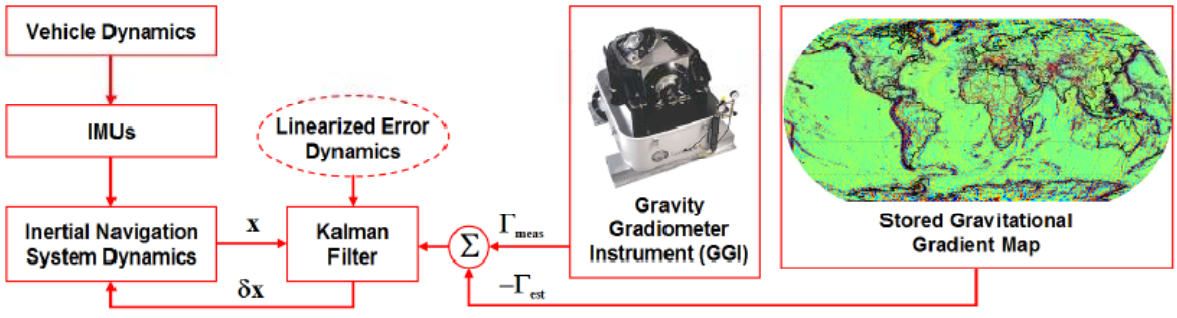


Figure 4: A Coupled INS and GGI Using Map Matching and a Kalman Filter[12]

Gravity Gradients

Newton's law of gravitation states that the attracting force between two masses occurs along a connecting line and has a magnitude, F, given by Equation 5, where G is the gravitational constant $(6.6742 \pm 0.0010) \cdot 10^{-11}$ $m^3 kg^{-1} s^{-2}$, m_1 and m_2 represent the mass of the two objects, and l represents the distance between the two masses.

$$F = G\frac{m_1 m_2}{l^2}$$

Equation 5: Newton's Law of Gravitation

Wellenhof and Moritz define a rectangular coordinate system similar to Figure 5. Then, using Newton's law of gravitation and setting the attracted mass, m_2, equal to a mass of unity, they produce the three components of the gravitational force vector, X, Y, and Z, shown in Equation 6. The direction of the force may also be defined by angles α, β, and γ, which measure rotation from the x, y, and z axes, respectively.[11]

$$X = -F\cos\alpha = -\frac{Gm_1}{l^2}\frac{x_2 - x_1}{l} = -Gm_1\frac{x_2 - x_1}{l^3}$$

$$Y = -F\cos\beta = -\frac{Gm_1}{l^2}\frac{y_2 - y_1}{l} = -Gm_1\frac{y_2 - y_1}{l^3}$$

$$Z = -F\cos\gamma = -\frac{Gm_1}{l^2}\frac{z_2 - z_1}{l} = -Gm_1\frac{z_2 - z_1}{l^3}$$

$$l = \sqrt{(x_2 - x_1)^2 + (y_2 - x_1)^2 + (z_2 - x_1)^2}$$

Equation 6: Gravitational Force Vector between Two Point Masses[11]

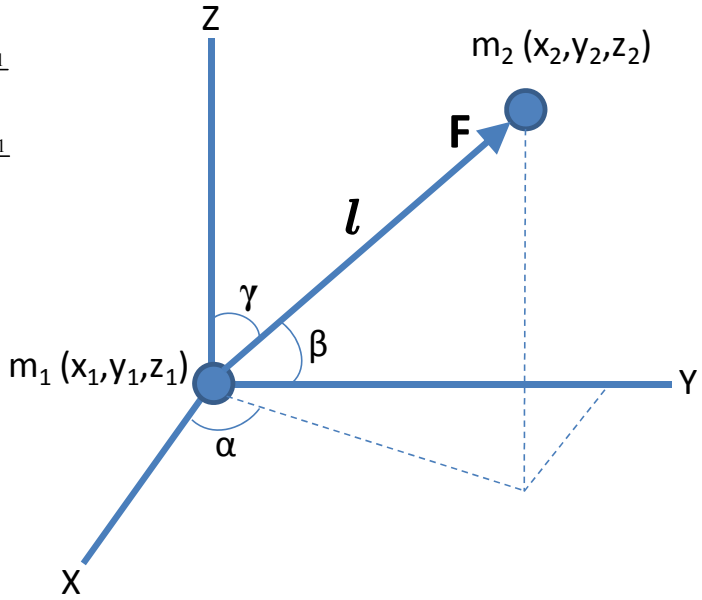


Figure 5: Components of the Gravitational Force[11]

It's interesting to note that if the Earth was assumed to be a point mass or perfect sphere with constant density, then, for a given altitude above the Earth's surface, the magnitude of F would be constant at all latitudes and longitudes and the direction F would always be in the down direction of the north, east, down reference frame. Since the Earth is not a perfect sphere with constant density, the terrain of the Earth is uneven, and the presence of other masses induces gravitational attractions, the gravitational vector varies between locations. Consequently, the spatial derivatives of the gravitational vector are not constant either. A more realistic model of the Earth would use an infinite number of infinitely small point masses to represent the Earth, where the total gravitational attraction would be the sum of the gravitational attractions induced by the presence of each infinitely small point mass. Thus, the total gravitational force would be found by summing the gravitational attraction force for all of the infinitely small point masses over the entire volume of the Earth. In Equation 7, ρ represents the density of the infinitely small point masses and dv represents the volume of the infinitely small point masses.

$$X = -G\iiint_{v} \frac{x_2 - x_1}{l^3}\rho dv$$

$$Y = -G\iiint_{v} \frac{y_2 - y_1}{l^3}\rho dv$$

$$Z = -G\iiint_{v} \frac{z_2 - z_1}{l^3}\rho dv$$

Equation 7: Gravitational Force Vector with an Infinite Number of Point Masses[11]

Due to the randomness of the Earth's terrain and density distributions throughout its volume, the gravitational force vector will also be spatially random. Consequently, the gravitational potential, φ, and gravitational gradients, Γ, will also be spatially random. Richeson provides a concise representation of gravitational potential, which is a scalar function, whose first and second spatial derivatives give the gravitational vector and gradients. The gradients are conveniently presented as a second order tensor with nine components. The following three equations use the north (N), east (E), down (D) reference frame, **g** to represent the gravitational vector, Γ to represent the gradients, and r and r' to represent the locations of the attracted and attracting masses, respectfully.[12]

$$\phi_g \equiv \iiint_v \frac{\rho(r')}{|r - r'|} dV$$

Equation 8: Gravitational Potential

$$g^n \equiv \nabla \phi_g = \begin{pmatrix} g_N \\ g_E \\ g_D \end{pmatrix}$$

Equation 9: Gravitational Force Vector in the North, East, Down Reference Frame

$$\Gamma^n \equiv \nabla \nabla^T \phi_g = \begin{pmatrix} \Gamma_{NN} & \Gamma_{NE} & \Gamma_{ND} \\ \Gamma_{NE} & \Gamma_{EE} & \Gamma_{ED} \\ \Gamma_{ND} & \Gamma_{ED} & \Gamma_{DD} \end{pmatrix}$$

Equation 10: Gravitational Gradients

In the matrix of nine gravitational gradients, the first subscript denotes the direction of the gravitational vector, which changes with a given movement in the direction of the second subscript. For example, the term Γ_{NE} represents the change in gravitational force in the north direction for a given movement in the east direction. Since $g^n \equiv \nabla \phi_g$, the algebraic properties of the del operator infer that

$$\nabla \times g^n = \nabla \times (\nabla \phi_g) = 0$$

Equation 11: Gravitation as a Conservative Field

Expanding the terms results in a symmetric gravitational gradient matrix.

$$\frac{\partial g_N}{\partial E} = \frac{\partial g_E}{\partial N}$$
$$\frac{\partial g_E}{\partial D} = \frac{\partial g_D}{\partial E}$$
$$\frac{\partial g_D}{\partial N} = \frac{\partial g_N}{\partial D}$$

Equation 12: Symmetric Terms in the Gravitational Gradient Matrix[13]

Equation 12 represents the same symmetrical terms as seen in Equation 10. Furthermore, when the density of the Earth is assumed much greater than the atmosphere, the trace of the gravitational gradient matrix equals zero, in accordance with Laplace's equation applied to the gravitational potential.

$$\Gamma_{NN} + \Gamma_{EE} + \Gamma_{DD} = 0$$

Equation 13: Free Air Assumption Applied to Gravitational Gradients[12]

Considering the previous four equations, the nine component gravitational gradient matrix only includes five independent terms. In other words, measuring five components captures the full second order tensor, with the assumptions previously discussed.[14]

To prevent confusion, it's important to define the difference between *gravity gradients* and *gravitational gradients*. This paper uses the terms interchangeably, but traditional definitions make a clear distinction between the two terms. *Gravitational* refers to forces defined by Newton's law of gravitation, while *gravity* refers to the sum of gravitational and centrifugal forces. Centrifugal generally refer to the force experienced by an object due to the Earth's rotation. Finally, it's also important to note the units of gravitational gradients. Since gravitational gradients represent the spatial derivatives of gravitation, which has the familiar units of $\frac{m}{s^2}$ or $\frac{ft}{s^2}$, then dividing by a change of length (m or ft) in a given direction would mean gravitational gradients carry units of s^{-2}. Due to the small magnitudes experienced in geodesy and the contributions of Baron Roland von Eötvös to this field of study, gravitational gradients are often communicated in units of Eötvös (*Eö*), where $1\,E\ddot{o} = 10^{-9}\,s^{-2}$. One *Eö* is equivalent to the gravitational gradient induced by 10 grains (i.e. ≈10 milligrams) of sand 1 centimeter away. [12]

$$1\,E\ddot{o} = 10^{-9}\,s^{-2}$$

Equation 14: The Eötvös Unit of Measurement for Gravitational Gradients

Gravity Gradient Instruments

Baron Roland von Eötvös, a Hungarian physicist, made the concept of a gravity gradient instrument a success in 1890, when he developed and employed a torsion balance to measure small gravity gradients induced by a nearby concentrated mass. The torsion balance represented the gravity gradient by the amount of twist in a thin wire, which suspended a metal beam with a weight at each end. When different gravity forces acted on the weights, separated by a known distance, a rotational force acted on the beam and twisted the thin wire. At the time, Eötvös's torsion balance provided the first successful measurement of gravity gradients and did so at precise locations with great sensitivity.[15] In his own words, Eötvös described the torsion balance as follows:

Figure 6: Eötvös's Torsion Balance

> "The means I use is a simple, straight stick with masses attached to each end and encased in metal, so that it will not be disturbed by the movement of air or differences in temperature. All mass near or far has an attracting influence on the stick, but the fibre, from which it is hung, resists this effect and twists in the opposite direction, producing by its twisting the exact measurements of the forces imposed upon it. This is nothing but an adapted version of the Coulomb instrument. It is as simple as Hamlet's flute, if you know how to play it. Just as the musician can coax entrancing melodies from his instrument, so the physicist, with equal delight, can measure the finest variation in gravity. In this way we can examine the Earth's crust at depths that the eye cannot penetrate and the rig cannot reach."[16]

Eötvös's direct measurement of gravity gradients at precise locations with the torsion balance remained unrivaled for many years. While Eötvös achieved a precision of ±1 to 3 Eö, the differencing of gravity measurements from different locations could still only achieve a gravity gradient precision of ±10 Eö in 1979. The concept of differencing gravity measurements, measured with gravimeters at different locations, provides a fundamental illustration of how modern gravity gradient instruments work. As shown in Figure 7, when two accelerometers are aligned in the same direction and separated by a known distance, their measurements may be differenced and then divided by the separation distance to obtain a gravity gradient. Consistent with popular notation, the first subscript denotes the accelerometers alignment direction, while the second subscript denotes the direction the accelerometers are separated by a known distance.[17]

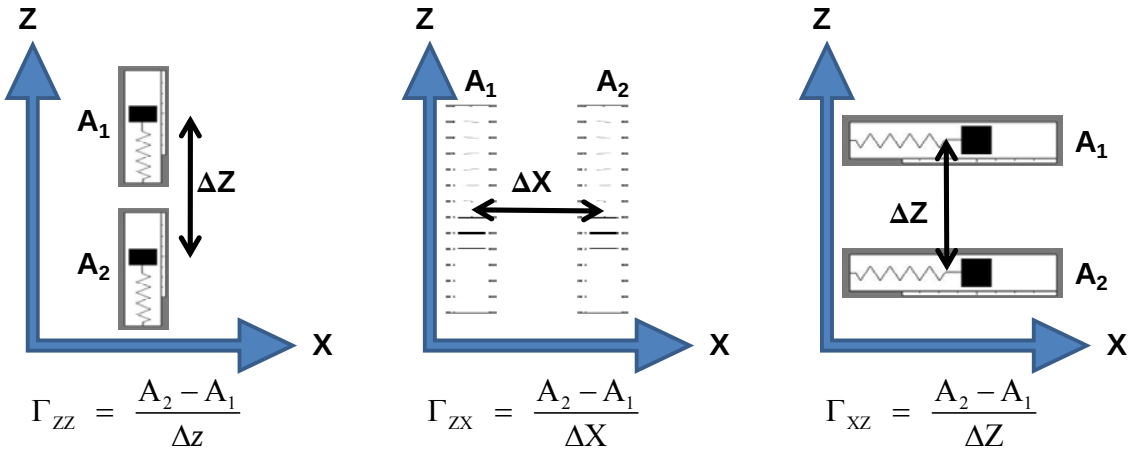


Figure 7: Gravity Gradients Measured with Accelerometers

Bell Aerospace took advantage of accelerometer differencing techniques that cancel out common forces and developed a GGI for use on moving vehicles. In general, translational vehicle dynamics cancel when two accelerometers attach to a rigid frame

and are differenced. Also, rotational dynamics might cancel or be considered small, if establishing a sufficiently small distance of separation between the accelerometers, while considering the magnitude of the rotational dynamics. Bell Aerospace's GGI employed a rotating accelerometer concept, with three gradiometers mounted on a gyro-stabilized platform. Each gradiometer included two accelerometers separated by a known distance. Precision reached 5 Eö and reinvigorated the application of GGIs, primarily because gravity gradient measurements from moving platforms presented the opportunity for rapid and convenient data collection over all kinds of terrain and even under water (e.g. onboard automobiles, aircraft, boats, and submarines).

While other types of GGIs exist today, the rotating accelerometer GGI stands as the only type of GGI successfully used in airborne surveys. Rogers provides an overview of the different types of GGIs currently in use and under development, including rotating accelerometer, superconducting, and atom interferometer GGIs. Based on his assessment of the current market, applications to airborne surveying, and GGIs under development, Rogers defines performance specifications for two generic GGIs in Table 3. The current GGI represents performance levels already demonstrated in tests, while the future represents an optimistic expectation of performance available within a decade.

Table 3: Approximate Performance Specifications of Current and Future GGIs

GGI	*NSD*	f_S	*RMS Noise*	f_C	*RMS Noise after Filtering*
Current	2.23 Eö√Hz	1 Hz	1.58 Eö	0.2 Hz	1.0 Eö
Future	0.223 Eö/√Hz	1 Hz	0.158 Eö	0.2 Hz	0.1 Eö

The noise spectral density (NSD) typically defines the noise level of GGIs with the assumption of zero mean and Gaussian distribution. NSD represents the power of GGI noise over a range of frequencies, measured in $\frac{E\ddot{o}}{\sqrt{Hz}}$. Equation 15 shows the calculation for RMS noise of a GGI in units of Eö, given the NSD and sampling frequency, f_s., in Hz.

$$RMSNoise(E\ddot{o}) = \sqrt{[NSD(\frac{E\ddot{o}}{\sqrt{Hz}})]^2 \times \frac{1}{2} f_s(Hz)}$$

Equation 15: RMS Noise Calculation for a GGI

Rogers also notes that GGI users commonly apply a low pass Butterworth filter to reduce noise. In Table 3, f_c represents the cutoff frequency of the low pass Butterworth filter and the final column gives the RMS Noise after filtering. Given a constant cutoff frequency, the spatial resolution of data from a GGI will increase as the vehicle's speed decreases. Alternatively, spatial resolution will decrease as speed increases. Increasing the cutoff frequency increases spatial resolution at higher speeds, but generally increases the noise passing through the filter. Additionally, if the cutoff frequency increases, higher signal frequencies pass through the filter, which might include frequencies higher than the Nyquist frequency, $f_{Nyquist}$. In this case, aliasing would occur, where the sampling rate is not sufficiently high and the ability to capture the signal's frequency spectrum is lost.[19]

$$f_{Nyquist} = \frac{1}{2} f_s$$

Equation 16: Nyquist Frequency

Gravity Gradient Maps and Surveys

In 1901, the head of the Hungarian geological survey, Hugo de Boeckh, persuaded Eötvös to bring his torsion balance onto a frozen lake. After taking measurements at 40 different locations on frozen Lake Balaton near Budapest, the team composed the world's first gravity gradient map. Pleasingly, the map matched the contours of the lake floor, which line and sinker measurements confirmed. Eötvös and the torsion balance gained instant fame in the geology community, including prospectors of valuable underground natural resources. Unfortunately, the contemporary difficulties of employing the torsion balance led to a lull in its application. These difficulties included the hardships of World War I, variations in temperature and wind that interfered with torsion balance measurements, sensitivity to nearby objects, and the skill required to interpret gravity gradient measurements.[15]

Even though geologists and prospectors preferred gravity maps over gravity gradient maps during this era of difficulty with application of the torsion balance, gravity gradient maps offer distinct advantages over gravity maps. While gravity maps illustrate up to three components (e.g. the force of gravity in the north, east, and up directions), gravity gradient maps include five independent components. Consequently and by their nature, gravity gradient maps provide clearer and more detailed information. Variations in gravity maps are relatively more subtle. Additionally, gravity gradient maps do not include noise from the erratic motion of the instruments, since the differencing technique between sensors eliminates these errors. This is great news for airborne surveys, which offer distinct advantages in data collection, but are subject to in-flight turbulence.[15]

The hypothetical consideration of the gravity gradients experienced when passing over a defined mass provides a more thorough understanding of the nature of gravity gradient maps. Rogers employed the closed form solutions for the five gravitational disturbance gradients, developed by Nagy, Papp, and Benedek[18], to illustrate the gravity gradient map that would result from passing over a rectangular prism with constant density. In this case, the gravity gradients were calculated and plotted on a plane 50 *m* above the rectangular prism, which Rogers defined with a density of 1.5 *g/cm3*, length of 50 *m*, width of 10 *m*, height of 6 *m*, and centered on a 250 *m* by 250 *m* grid.

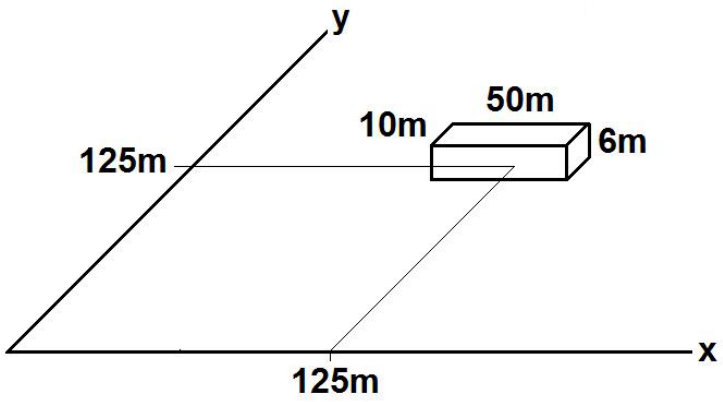


Figure 8: Hypothetical Prism[19]

Figure 9 shows the gravity gradient maps that Rogers produced in MATLAB, using the closed form gravitational disturbance gradient solutions for the hypothetical prism. Since the tensor is symmetric, only the upper right triangular portion of the matrix is presented. If portrayed in the NED reference frame, x, y, and z might be considered north, east, and down, respectively. These maps illustrate theoretical gravity gradients, T, and provide an

excellent illustration of the uniqueness that makes gravity gradients an excellent foundation for map matching navigation.[19]

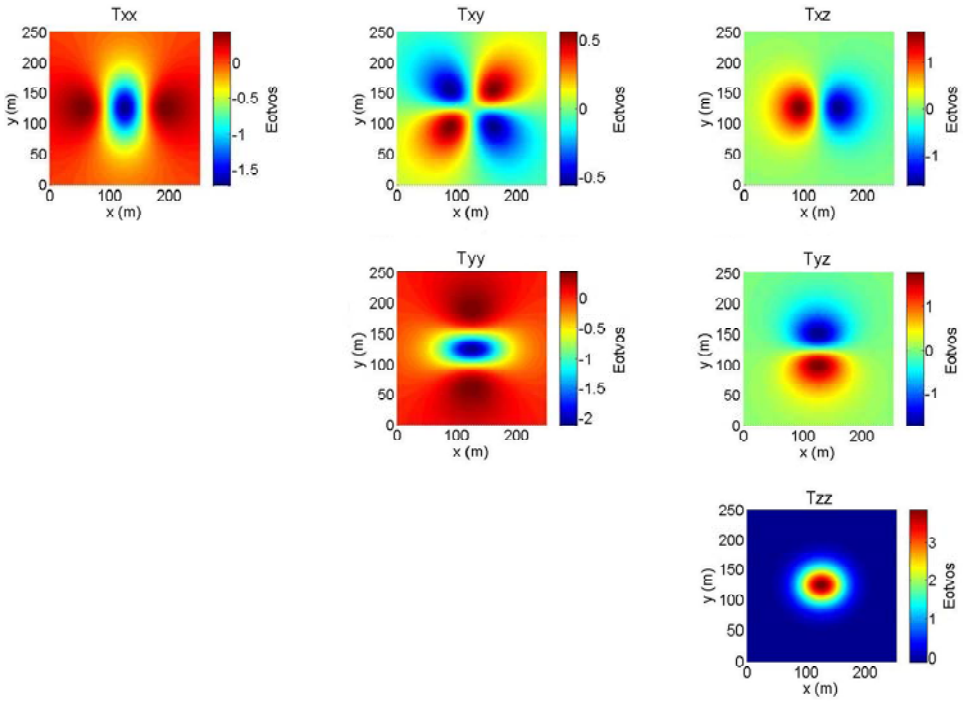


Figure 9: Gravity Gradient Map on a Plane 50 meters Above Hypothetical Prism[19]

Equation 5 shows that gravitational forces change inversely to the square of the distance between masses. Gravity gradients represent spatial derivatives of gravitational forces and thus change inversely to the cube of the distance. This applies to gravity gradient maps over hypothetical prisms as well as the Earth. Richeson uses the Earth Gravitational Model 1996 (EGM96) to show how gravity gradients change with altitude. Since the Earth and its terrain features dominate the gravitational forces and gradients in this scenario, variations in gravity gradients occur over mountainous regions like the Rocky and Andes Mountains and attenuate cubically as altitude increases. Richeson also

presents estimates of when terrain effects on gravity gradients can be ignored (i.e. when terrain contribution to gravity gradients is less than GGI noise levels).

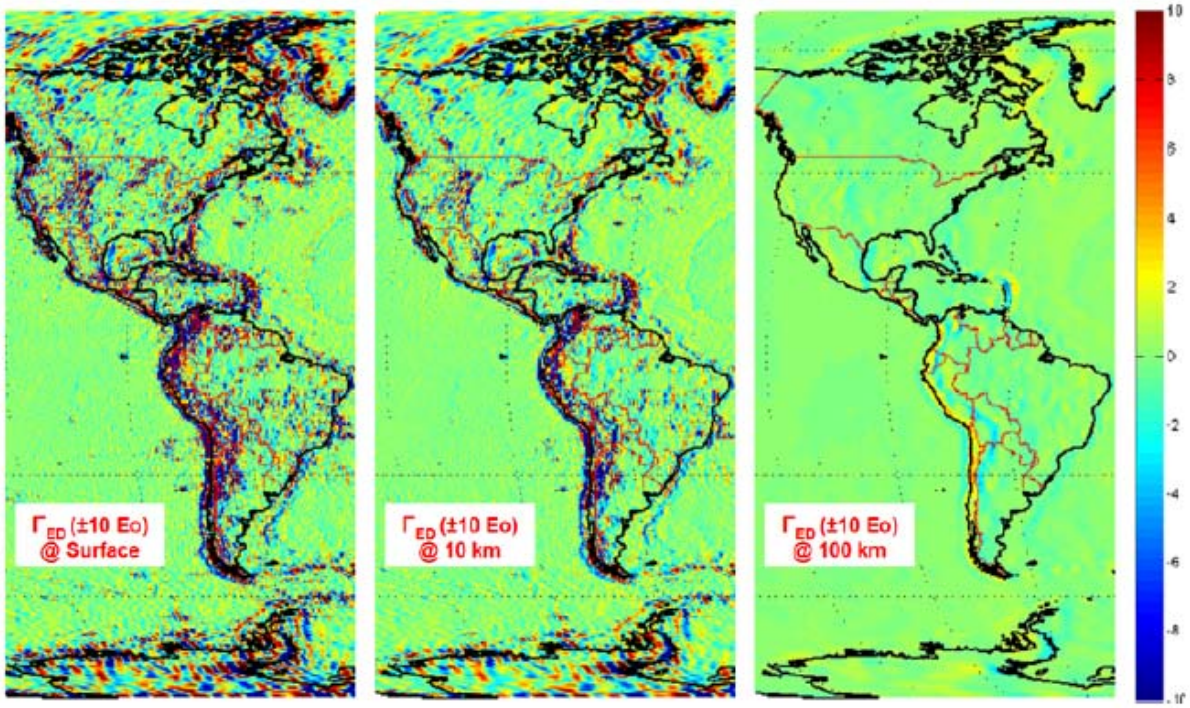


Figure 10: East-Down Gravitational Gradient at Three Altitudes[12]

The following page shows maps over Earth's surface for six of the nine gravity gradient components, excluding symmetric terms, with color scales in Eö units. Note that unlike TRN and visual observations, gravity gradients provide contrasts over bodies of water. Additionally, while INS vertical errors might increase due to significant changes in gravitational forces (e.g. over mountains), these changes provide more contrast on gravity gradient maps, thus improving the potential for more accurate navigation.

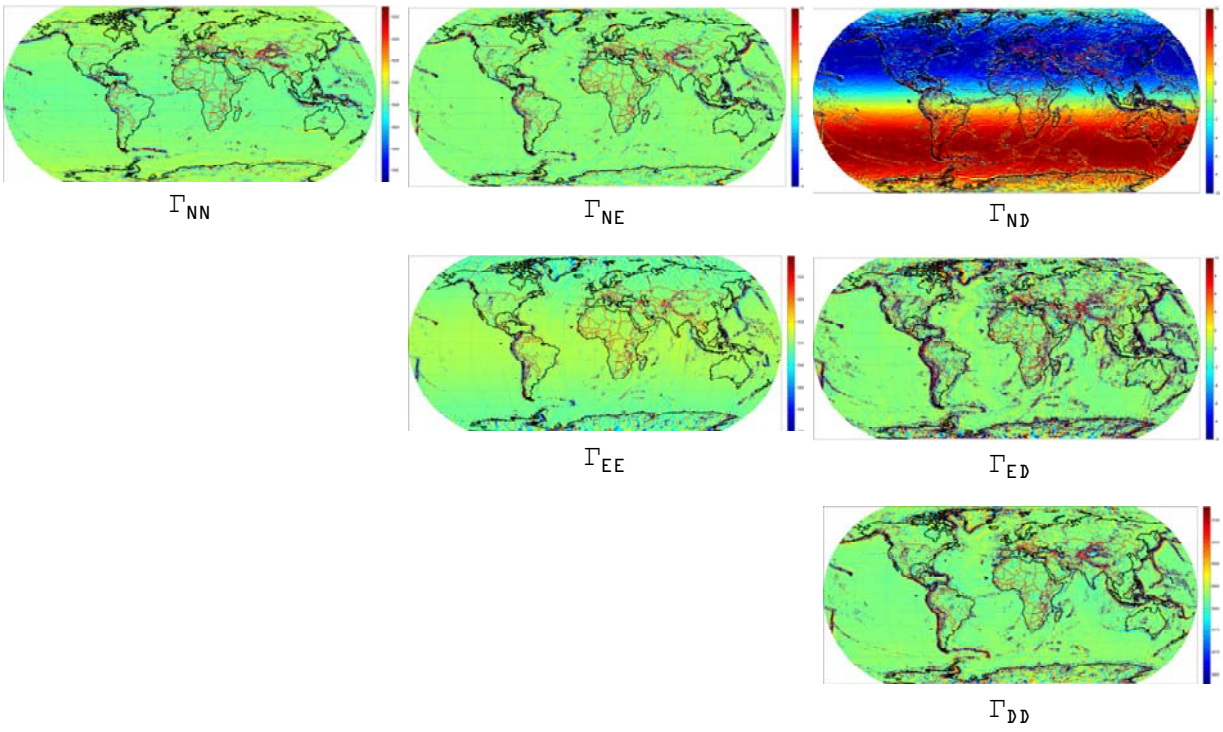


Figure 11: Map of Gravity Gradients on Earth's Surface[12]

While the gravity gradient matrix includes five independent terms or sources for information, the third column appears to provide the most contrast (Γ_{ND}, $\Gamma_{ED,\text{ and }}\Gamma_{DD}$), suggesting the most potential for accurate navigation solutions. These three terms represent the gravity gradients in the three reference frame directions, given a movement in the down (i.e. vertical) direction. Richeson notes the finite resolution of the EGM96 model means that realistic gravity gradients at low altitudes are most likely larger than they appear on his maps, since sharp terrain effects might be masked. If the resolution of the map increased, then more information would be available for navigation applications. However, the range, sensitivity, and noise of the GGI employed would also affect navigation performance.

Since contrasts on gravity gradient maps form the foundation for building navigation information, the standard deviation of gravity gradients over a given area on the map provide a quantifiable measure of its value. The standard deviation also reflects a measure of how much weight a Kalman filter might give to navigation information derived from gravity gradients and map matching. For example, an aircraft flying in a region with very small standard deviations in the gravity gradients, would have a lower probability of gaining valuable information from gravity gradiometry and map matching. On the other hand, flying in a region with large standard deviations would result in a higher probability of gaining valuable information. As expected, the largest standard deviations in Richeson's Γ_{DD} map occur in the mountainous regions of the world, in addition to some locations over water.[12]

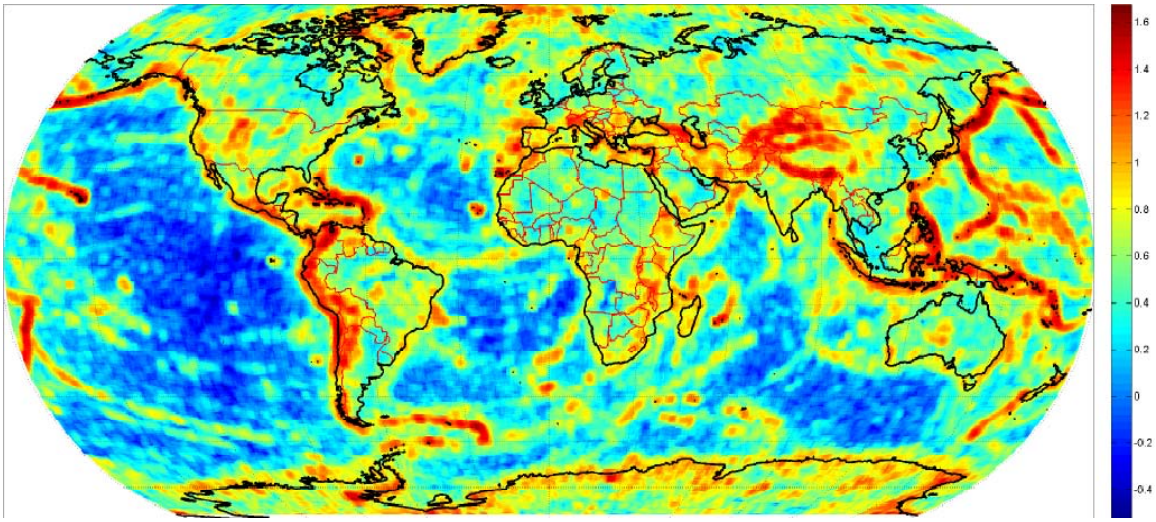


Figure 12: Γ_{DD} Standard Deviation on Earth's Surface [$\log_{10}$(Eö)][12]

Even though many applications focus on the T_{zz} component of the gravity gradient tensor, probably due to its interpretive ease and contrasts, the T_{xz} and T_{yz} components arguably provide just as much, if not more information, than the T_{zz} component alone. Veryaskin and McRae showed that T_{xz} and T_{yz} produced the same information as T_{zzz}, which represents the partial of the T_{zz} component of the gravity gradient tensor with respect to the z coordinate. Knowing that the trace of the gravity gradient tensor equals zero (i.e. $T_{xx} + T_{yy} + T_{zz} = 0$), the partial of the entire expression with respect to z yields Equation 17 after reordering the derivatives. The authors caution that noise increases when using this technique.

$$\frac{\partial T_{xz}}{\partial x} + \frac{\partial T_{xz}}{\partial x} = -T_{zzz}$$

Equation 17: The Third Vertical Derivative of Gravitational Potential

Additionally, Veryaskin and McRae proposed that T_{xz} and T_{yz} could be treated as two orthogonal components of a vector, whose magnitude is single valued and independent of orientation in the horizontal plane. This single vector modeling technique, however, still receives contributions from angular rates and accelerations, but demands less accuracy in the magnitude of the individual gravity gradient components, assuming the single vector magnitude remains the same.

$$T = \sqrt{T_{xz}^{\ 2} + T_{yz}^{\ 2}}$$

Equation 18: Single Vector Magnitude of T_{xz} and T_{yz}

In practice, Veryaskin and McRae propose that the single vector modeling technique could be used for high-altitude, large-scale surveys, and then the third vertical derivative technique for more refined surveys at lower altitudes.[20]

Mickus and Hinojosa also showed that Fast Fourier Transforms (FFT) enable calculation of the complete gravity gradient tensor from data only on the vertical component of gravity. The basic expressions used in their analysis were derived from the assumption that the gravitational potential, φ, is a scalar function of the x, y, and z coordinates and satisfies Laplace's equation, $\nabla^2\varphi = 0$. As such, the Fourier Transform of the gravitational potential, Φ, is a function of the wave number vector, $[k_x, k_y, k_z]$.

$$(k_x^2 + k_y^2 + k_z^2)\,\Phi(\mathbf{k}) = 0$$

Equation 19: Fourier Transform of Gravitational Potential

With further knowledge that the curl of the gravitational field is zero, Mickus and Hinojosa derive the wave number matrix, $K(\mathbf{k})$, and the final expression for the gravity gradient tensor, Γ_{ij}, where *i* and *j* represent the x, y, or z coordinates and $G_z(\mathbf{k})$ represents the Fourier Transform of the vertical component of the gravitational vector.

$$\left[K(k)\right] = \begin{pmatrix} \frac{-k_x^{\,2}}{|k|} & \frac{-k_x k_y}{|k|} & -ik_x \\ \frac{-k_x k_y}{|k|} & \frac{-k_y^{\,2}}{|k|} & -ik_y \\ -ik_x & -ik_y & |k| \end{pmatrix}$$

Equation 20: Wave Number Matrix for Gravitational Potential Fourier Transform

$$\Gamma_{ij} = \mathcal{F}^{-1}\left\{\left[K(k)\right]G_z(k)\right\}$$

Equation 21: Gravity Gradients using Gravitational Potential Fourier Transform

The application of FFT to calculate the gravity gradients from data on the vertical component of gravity induced error. The RMS errors ranged from a minimum of 0.3 *Eö* for the g_{xx} component to 3.3 *Eö* for the g_{zy} component. When applying this technique to data on the vertical component of gravity from a region in southwestern Oklahoma and comparing it to gravity gradient data measured in an airborne survey by the United States Air Force's Gravity Gradient Survey System (GGSS), trends generally matched, but errors were difficult to analyze due to lack of quality in the measured data.

Map Matching Algorithms

In the context of this research, map matching algorithms use gravity gradient measurements to locate an aircraft's position on a gravity gradient map. Many techniques exist for accomplishing this function, and each technique possesses strengths and weaknesses. The nature and robustness of the map matching algorithm, as well as characteristics of the GGI signal and the gravity gradient map, affect the ability to make a match, the precision, and the accuracy of the match. Figure 4 and Figure 13 present two perspectives of the architecture surrounding a map matching algorithm. In Figure 4, the summation symbol, Σ, represents the map matching algorithm. Figure 13 includes an illustration of the assumption that information from the INS will be available to assist the map matching algorithm. This should make map matching easier, but might require a more robust algorithm when information from the INS is not available, especially

considering the scenario where the INS is initialized and cannot provide an initial position estimate to the map matching algorithm.

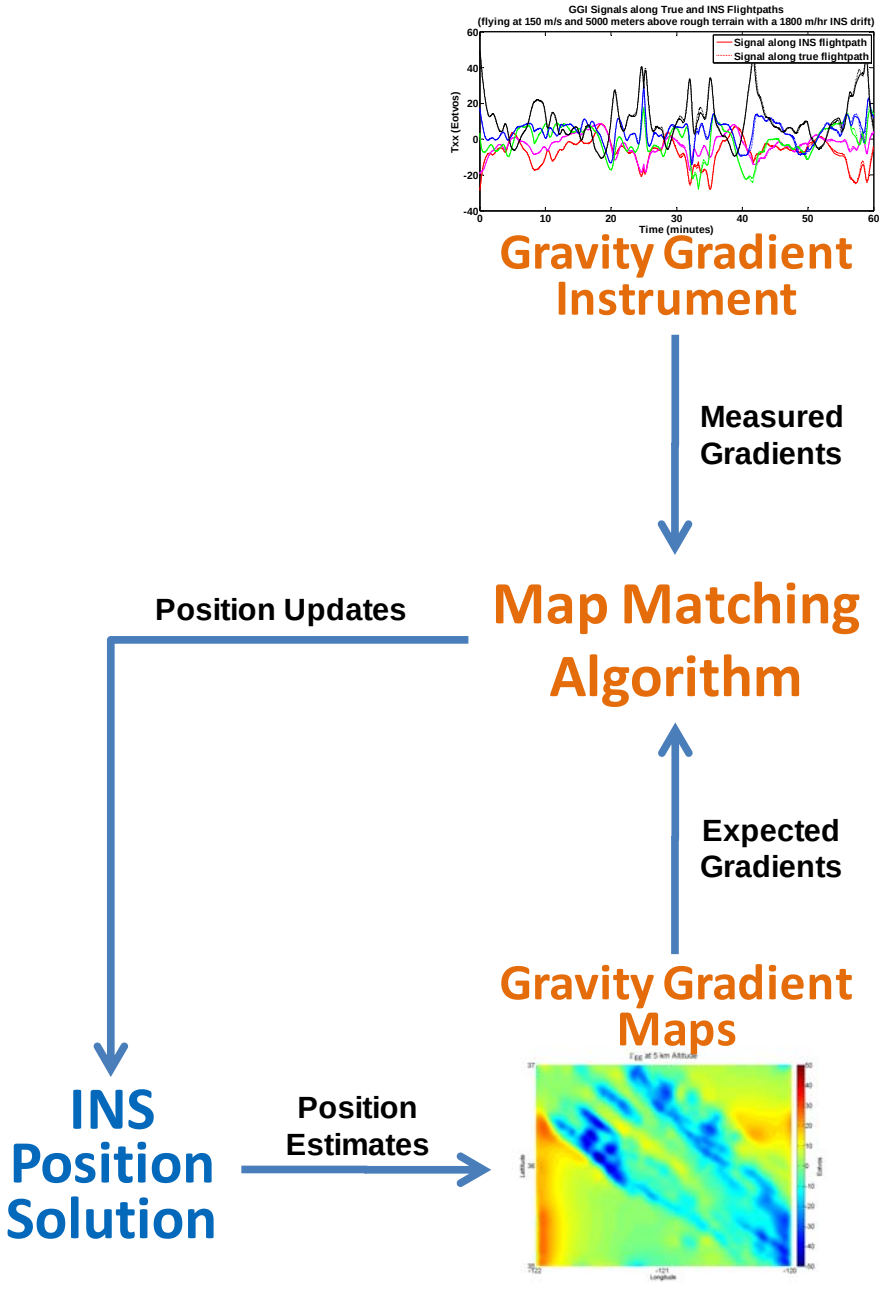


Figure 13: Map Matching Algorithm as Part of an INS/GGI Navigation System

Many concepts exist for building map matching algorithms, and this paper presents some of the concepts that could be applied to this research. First, a map matching algorithm may use one or more measurements to attempt a match to the map. Single beam measuring describes the measurement of single points along a path, while multibeam measuring describes the capture of many measurements simultaneously while traveling along a path. Nygren shows that multibeam measuring improves the accuracy and robustness of TRN. [33] Greenfeld distinguishes between map matching algorithms that only utilize geometric information and those that are *topological*. Topological refers to matches "done in context and in relationship to the previously established matches" (p.4). In the context of matching GPS observations to a digital map, Greenfeld argues that topological solutions are more likely to be correct than solutions based only on geometry.[21]

Gallagher provides a window into the diverse art of graph-based pattern matching. If GGI measurements along a path are visualized graphically, then a gravity gradient map may be perceived as a database of graphs. In this case, Gallagher's research presents several methods for building map matching algorithms, including structural matching, such as vertex and geometry matching, structural mining, semantic matching, and similarity based matching. [22]

While some might think of landmarks as visual references, Dedeoglu and Sukhatme apply the concept to the topological maps collected by autonomous robots. In the case of robots, collaborative mapping occurs when two different robots identify the same landmarks on their maps, thus enabling a map match. In the case of gravity gradiometry, one might think of the gravity gradients induced by unique shapes and

densities of masses as landmarks. Thus, the measurement of unique gravity gradients might act as a landmark on a map.[23] Some might interpret this concept as feature matching, where a unique feature of a map is found in the measured data. Easily recognizable map features, or gravity gradients due to unique shapes and densities of masses, might be preloaded in a database to provide adequate coverage of an area and minimize data processing burdens.[24]

The *coverage measure* focuses on the similarity between line segments that overlap. Although the authors primarily apply this method to orthogonal line segments encountered as robots map the interior of buildings, the idea of a coverage measure could apply to the comparison of segments of gravity gradient data to a map. After all, the realities of gravity gradient data processing include discrete sampling, possible temporal lapses in usable data, and maybe even discontinuities in the gravity gradient map. Application of the coverage measure to gravity gradient map matching might include an algorithm that compares measured data segments to map data segments, thus arriving at the position and/or bias that maximizes coverage (i.e. position location).[25]

Map matching algorithms that focus on Terrain Referenced Navigation (TRN) present a myriad of methods, including those previously discussed. One of the most popular methods, Terrain Contour Matching (TERCOM), determines position by calculating the mean absolute distance (MAD) between the expected and measured values along the navigated path. TERCOM determines the position by finding the path on the map whose values best correlate with the measured values (i.e. minimum MAD). Although this method makes the accuracy of the position solution difficult to determine, TERCOM's use in cruise missile navigation testifies to its reliability. Terrain Profile

Matching (TERPROM) exploited TERCOM for initialization of navigation solutions, but used Sandia Inertial Terrain Aided Navigation (SITAN) for tracking, which employed extended Kalman filters and matched local terrain gradients. Hagen identifies SITAN as more suitable for topographies with clearly defined gradients and TERCOM more suitable for rough topography.[26] In a paper about TerrLab, Hagen further describes the Point Mass Filter (PMF) and Particle Filter map matching algorithms. The Norwegian Defence Research Establishment developed TerrLab to assess the performance and robustness of TRN algorithms. TerrLab supports TRN aids loosely integrated with an INS, where *loosely integrated* refers to a navigation system that feeds the TRN position solution directly into the INS. According to Hagen, PMF uses a non-linear, Bayesian estimate of the state vector's probability density function (PDF). Particle filters also use a non-linear Bayesian estimator, but select particles to represent the PDF and propagate them forward in time according to the system's dynamic model.[27]

Archibald, Di Massa, and Dumrongchai devised map matching algorithms for the specific purpose of matching gravity gradients to a map. Archibald provided position updates to an INS by using digital terrain elevation data and a nearest neighbor neural network pattern match to determine a location on a map.[28] Di Massa briefly discussed similarity and dissimilarity parameters for matching, such as the cosine coefficient, correlation cooefficient, Canberra Metric, and Bray-Curtis Coefficient, but ultimately chose the MAD for her work. Di Massa also presented details on coarse-to-fine search methods, which attempt to reduce the computational burden of matching gravity gradients to large maps by starting with coarse, down-sampled data and identifying progressively finer areas until achieving a map match at the target resolution. With this

method, Di Massa emphasizes that a coarse-to-fine map match might not be the same solution as the solution obtained using an exhaustive search (i.e. there's a risk in missing a better match).[29] Dumrongchai provided a robust analysis of how matched filters can handle noise and detect small mass anomalies near the surface of earth. Although Dumrongchai identified the vertical gravity gradient as capable for detection, a matched filter that utilizes six components of the gravity gradient tensor provided improved results.[30]

A recent AFIT master's thesis, pertaining to the matching of magnetic field measurements to a map[31], Storms employed terrain navigation concepts published by Nygren.[32] In this method, a system model is defined where x_t represent the position at time t, u_t represents the distance traveled during that time step according to the INS, and v_t represents the error in the distance provided by the INS. The measurement, y_t, relates to the expected measurement according to the map, $h(x_t)$, and the combined error presented by the measurement and map, e_t, which is assumed independent, white, and Gaussian.

$$x_{t+1} = x_t + u_t + v_t$$
$$y_t = h(x_t) + e_t$$

Equation 22: System Model for Correlator Method of TRN

After application of Bayes' rule (Equation 23) and the establishment of a function that gives the likelihood of a measurement given a position (Equation 24), the posterior PDF (i.e. the PDF after inclusion of the position measurement) may be found (Equation 25).

The PDF is represented by p, while N represents the number of measurements considered in the PDF, C_e represents the measurement error covariance matrix, and $p[y_{t+1}]$ is treated as a normalizing constant. In essence, the posterior PDF represents a combination of the old PDF propagated forward in time and the PDF associated with the likelihood of obtaining the measurement. Nygren presents an illustration of this method in Figure 14 on page 45 and recommends the finite difference filter as a robust, accurate, and easy to implement method for calculating the posterior PDF when there are three or less states.[32,33]

$$P[B/A] = \frac{P[A \cap B]}{P[A]} = \frac{P[A/B]P[B]}{P[A]}$$

Equation 23: Bayes' Rule

$$L(y_t/x_t) = \frac{1}{\sqrt{(2\pi)^N \det(C_e)}} e^{-\frac{1}{2}[y_t - h(x_t)]^T C_e^{-1}[y_t - h(x_t)]}$$

Equation 24: Likelihood Function

$$p[x_{t+1}/y_{t+1}] = \frac{L[y_{t+1}/x_{t+1}]\, p[x_{t+1}/y_t]}{p[y_{t+1}]}$$

Equation 25: Posterior Probability Density Function

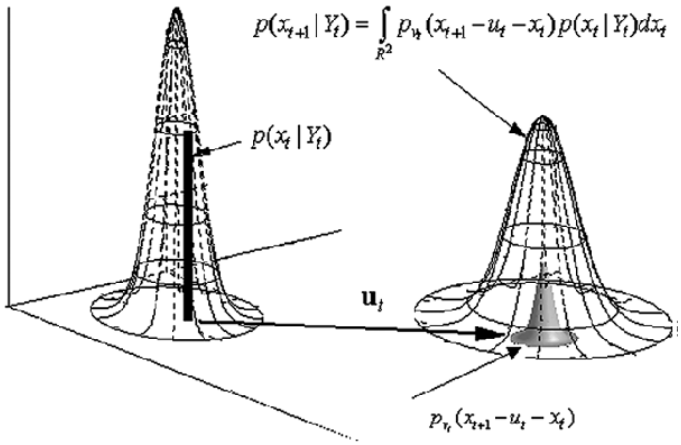


Figure 14: Propagation of Probability Density Function for Vehicle Position[33]

Bergman presented the fundamentals of applying Bayes Rule to TRN in a 1997 paper, including the use of point mass filters, but also discussed the gradient approach. In the case of a gravity gradient map, the gradient approach refers to the gradient of gravity gradients, which would be a third order tensor with 81 components. Bergman points out that the gradient approach removes bias from the estimation problem, but introduces higher noise levels.[34]

III. Methodology

Modeling and simulation provide the foundation of information for reaching the research objectives. A computer program takes user inputs and applies models of an aircraft, INS, GGI, and Earth's gravity gradients to calculate GAME performance. GAME performance calculations use standard performance measures, such as the root mean square (RMS) and 50th percentile circular error probable (CEP) of the position errors, plus two metrics unique to this research. A map matching algorithm applies GGI sensor data and a map of Earth's gravity gradients to calculate position solutions. A Kalman filter uses inputs from the INS and map matching algorithm to arrive at the best position solutions, which this paper refers to as *GAME solutions*. This paper also refers to the position solutions based only on INS information as *INS solutions* and solutions based only on gravity gradiometry and map matching as *GGI solutions*.

Computer Program

The computer program in Appendix A performs the simulations for this paper. All simulations run in MATLAB R2008b on a personal computer system running Microsoft Windows XP Professional with a Xeon X5482 processor, Intel 5400 chipset, and four gigabytes of random access memory. The operating system's 3GB switch gives MATLAB enough virtual memory to create the gravity gradient maps with a modified version of the computer program written by Rogers for his master's thesis in 2009.

The computer program begins by requesting the following inputs from the user: terrain, altitude, velocity, INS drift rate, GGI data rate, GAME position update rate, GGI sensor noise level, gravity gradient map noise level, simulated map resolution, amount of

map interpolation to be used in the map matching algorithm, duration of flight, and a filename for the results to be recorded. The aircraft flightpath and starting position, the gravity gradient maps used as the truth data, and the time step of the computer algorithm are hard coded in the computer program, but can be easily changed.

Next, the computer program loads gravity gradient maps based on the user's inputs and initializes variables. Afterwards, the computer enters a loop for the requested duration of flight, unless certain circumstances cause the program to terminate (e.g. the aircraft or INS drifts off the map). Each loop represents a time step in the simulation, which was hard coded at 1 hertz, but can be easily changed. Inside each loop, the computer algorithm calculates the true position of the aircraft and records the GGI signal at the true location with the user-specified noise added. Then, the computer program calculates the INS and GGI position solutions and uncertainties, which feed into a Kalman filter. The Kalman filter calculates a best position solution and uncertainty, which it feeds back to the INS and map matching algorithm. The loop also calculates the position error of the GAME and GGI solutions.

Finally, the computer program calculates the performance metrics, writes the inputs and results to a file, and provides five key plots: GAME position error versus time, GGI position error versus time, GGI signals along the true and INS flightpath versus time, latitude and longitude of the true and INS positions versus time, and a bird's eye view of the aircraft's flightpath.

The following figure provides a conceptual representation of the computer program, and the following sections discuss the specifics of the computer program's core models and algorithms.

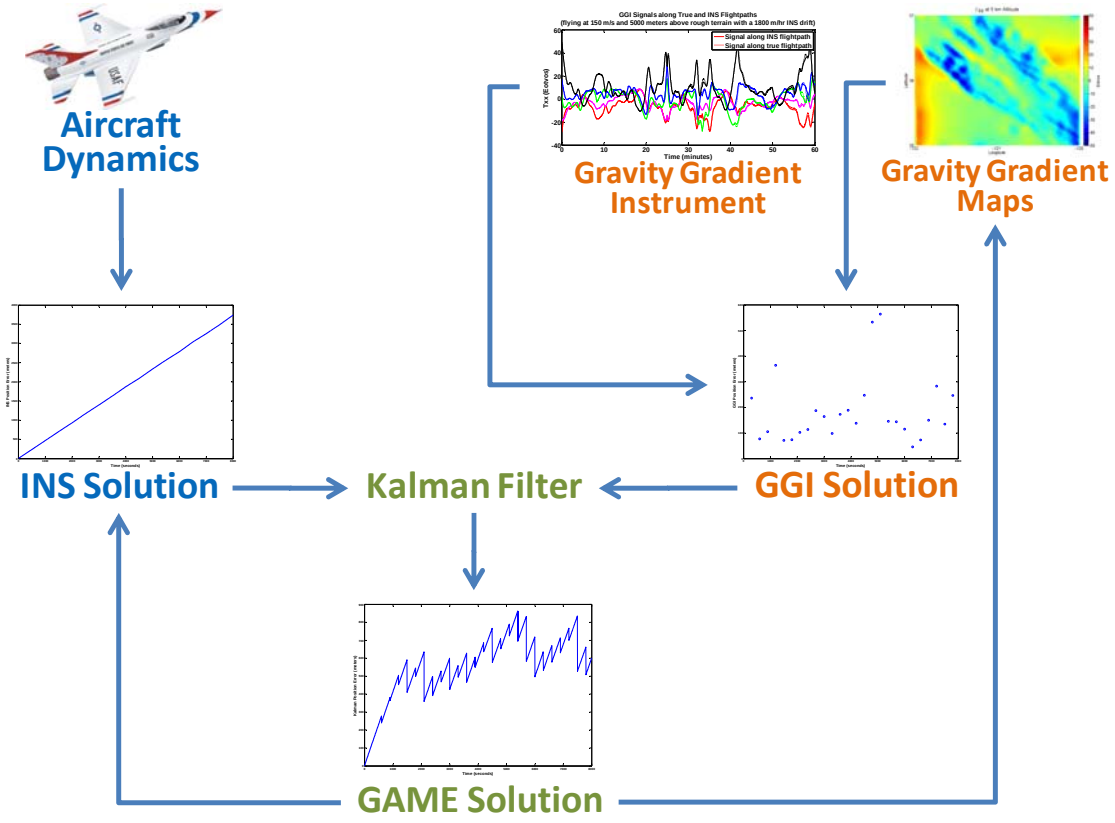


Figure 15: The Concept of the GAME

Aircraft Model

The aircraft flies at a constant velocity and altitude for the duration of flight, all given by the user. The model does not include angles of attack and sideslip, translational and rotational accelerations, and roll, pitch, and yaw positions, rates, and accelerations. This effectively means the model only calculates the aircraft's true position. The computer program also treats the flightpath and starting point as constants, although the user may change them. Simulations in this paper use the same flightpath and starting point, so comparisons of results include the same set of data points from the maps. Comparisons include the exact same data points from the set when the simulations fly the same distance, which ensures the effects of terrain can be isolated from other variables.

The simulations in this paper fly an 8-segmented star pattern to ensure the aircraft stays on the modeled maps, flies in a variety of directions, flies over a variety of terrains from different approach angles, and remains on the map for flights of great distances. Since each repetition of the star pattern moves slightly west of the previous star to maximize flight over a variety of terrains, the computer program terminates and provides notice if the aircraft flies too close to the edge of the map. This ensures the program does not crash and edge effects of the map do not significantly influence the results.

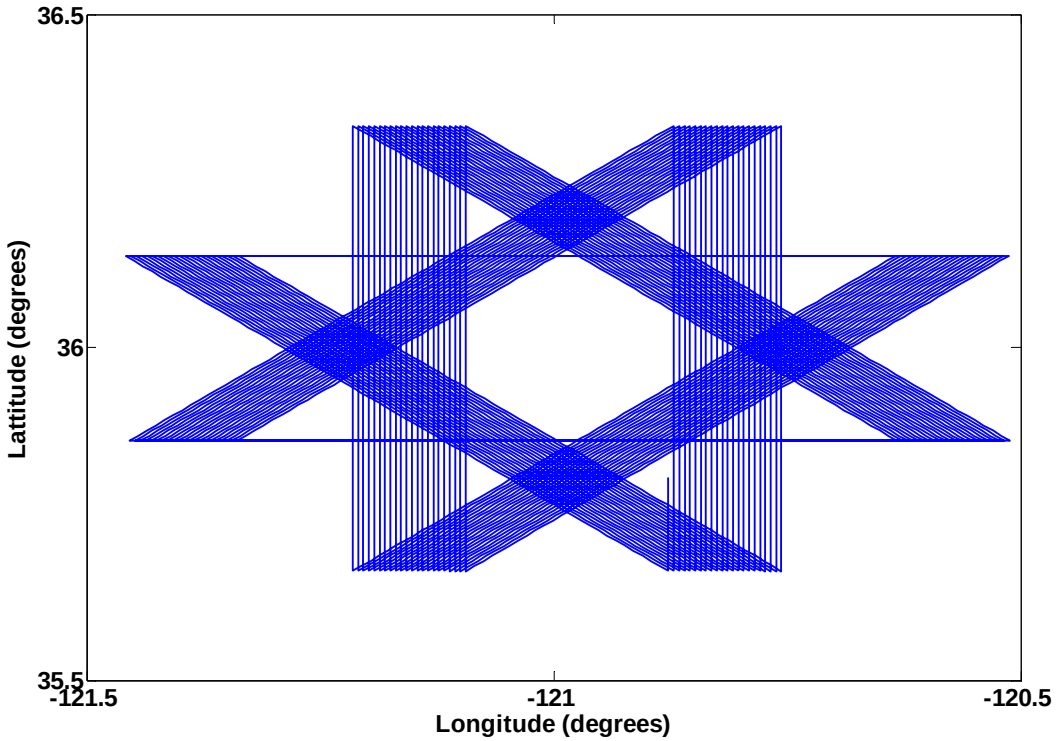


Figure 16: Simulated Aircraft Flightpath - An 8-Segmented Star

Inertial Navigation System Model

The computer program uses a simple INS model that only includes the INS position solution and uncertainty. The INS position solution drifts away from the aircraft's true position at a rate equal to the INS drift rate specified by the user. To ensure the effects of INS drift on GAME performance can be isolated from other variables, the drift always occurs in the southeast direction. The user can easily change the direction and magnitude of the drift in the computer program. INS drift does not occur in the vertical direction.

The uncertainty of the INS position solution starts at zero and increases at a rate equal to the INS drift rate specified by the user, but converted from a 50th percentile CEP to a variance for the uncertainty matrix. While the INS uncertainty increases at a constant rate in the north and east directions, position and uncertainty updates from the Kalman filter result in corrections to the INS position solution and uncertainty, which means the uncertainty of the INS will generally not be the same in the north and east directions.

The computer program records the INS position solutions, position errors, and gravity gradients along its flightpath for use in the analysis. The computer program terminates and provides notice if the INS position solution drifts off the map, in order to prevent the code from crashing. This simple INS model adequately covers the scope of this research effort, provides an opportunity to understand the effects of INS drift rates on GAME performance, and ensures that the INS behaves in a consistent manner, so the effects of other variables can be isolated during comparisons of results.

Gravity Gradient Instrument Model

The GGI model records five independent components of the gravity gradient tensor (T_{xx}, T_{xy}, T_{xz}, T_{yz}, and T_{zz}) at a rate specified by the user. This paper also refers to these components as Γ_{EE}, Γ_{NE}, Γ_{ED}, Γ_{ND}, and Γ_{DD}, respectively, in the north, east, and down reference frame. The model takes the gravity gradients from the modeled maps, which the computer program treats as truth data, and interpolates to arrive at a value based on the aircraft's true position. The model then takes the values and adds random noise based on the user's inputs. The computer program assumes information from the GGI is accurately time stamped and in the exact reference frame, or errors are compensated and within the simulated noise levels.

Gravity Gradient Maps

A computer program written by Captain Marshall Rogers, in support of his 2009 master's degree thesis at the Air Force Institute of Technology, provided the genesis of the modeled gravity gradient maps. After some minor modifications, Roger's computer program generated gravity gradient maps specifically for this research effort. These maps possess a resolution of 3 arcseconds and are derived from Earth Gravitational Model 1996 (EGM96) and Level 1 Digital Terrain Elevation Data (DTED). Roger's paper provides the details of the derivation.[19] Although this paper treats the maps as truth data, imperfect models and computations produced the maps. Thus, the modeled maps do not perfectly represent gravity gradients in the real world, but provide realistic trends and magnitudes.

The database includes map sets for two different areas in the United States, which provides an opportunity to learn how terrain affects GAME performance. Both areas measure 2 degrees latitude by 2 degrees longitude, about 222 by 180 kilometers. The first map set covers an area along the Pacific Coast of California between Sacramento and Los Angeles. This area provides significant variations in terrain mass, including ocean, flatlands, and mountains, between 1,800 meters above sea level and slightly below. The second map covers an area near the Mississippi River in Western Tennessee. This area provides small variations in terrain height between 0 and 250 meters. This paper refers to the first map set as *rough terrain* and the second as *smooth terrain*.

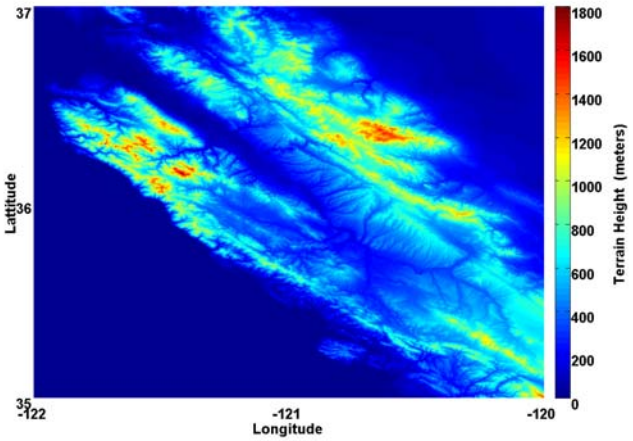


Figure 17: Rough Terrain

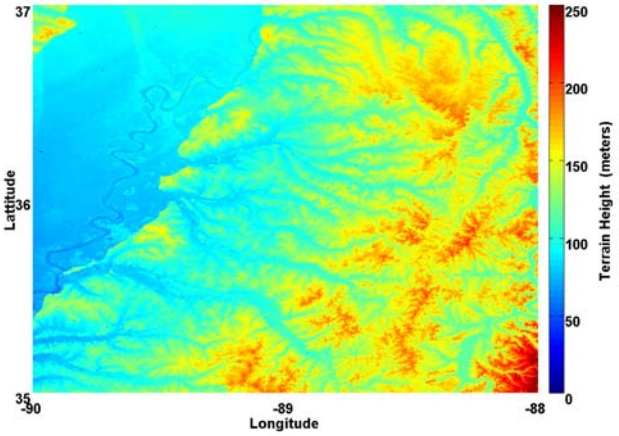


Figure 18: Smooth Terrain

The map sets for each area also include maps for 5 independent components of the gravity gradient tensor and at six different altitudes (5, 10, 15, 20, 25, and 30 kilometers above the average terrain height). The following figure of Γ_{DD} over rough terrain illustrates how the modeled gravity gradient maps attenuate as altitude increases.

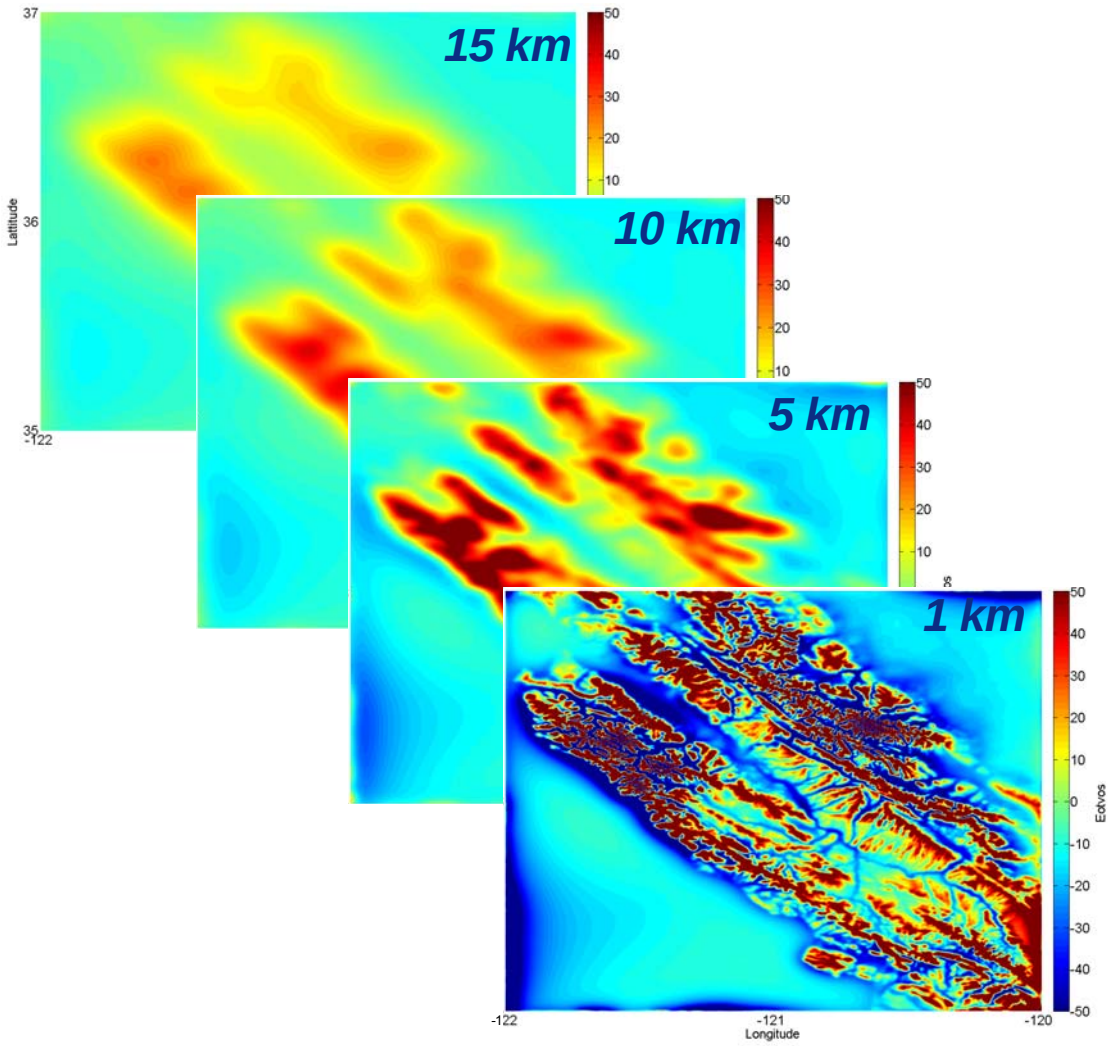


Figure 19: Γ_{DD} Attenuating as Altitude Increases over Rough Terrain

Figure 20 shows an example of the maps for the five independent components of the gravity gradient tensor used in this research. The component is labeled in the bottom left corner of each map, and all the maps are for an altitude of 5 kilometers.

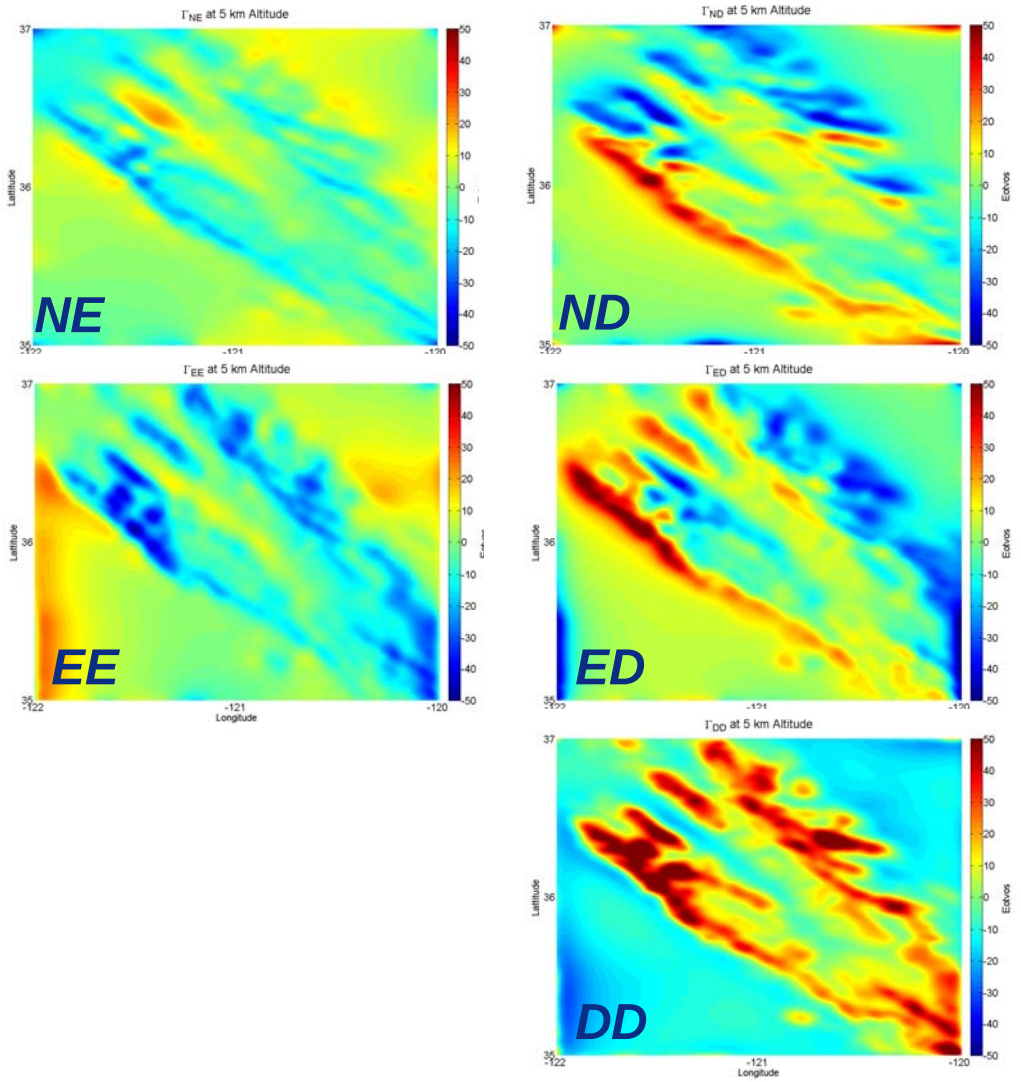


Figure 20: Five Components of the Gravity Gradient Tensor over Rough Terrain

All the maps were stored in a database prior to the flight simulations to ensure instant availability of gravity gradient information to the map matching algorithm. Processing times would be unacceptably long if the computer program used the EGM96 models and DTED information in real time to calculate the expected gravity gradients. At high altitudes, the maps could neglect the effects of terrain[12], thus reducing processing times and making real-time calculations of expected gravity gradients significantly faster. However, the maps in this research effort always include terrain effects.

Map Matching Algorithm

The map matching algorithm uses information from the GGI sensor and database of maps to determine GGI position solutions at a frequency specified by the user. The likelihood function discussed on page 44 provides the heart of the specific method chosen for this algorithm. This method inherently relies on the assumption that measured gravity gradients best match the expected gravity gradients at a unique location. While patterns of gravity gradients might be considered unique, like fingerprints, multiple locations with the same gravity gradient magnitudes should be expected. However, when five discrete measurements at a single location are compared within a smaller region of the world, the probability of finding multiple locations with the same gravity gradient magnitudes significantly decreases and makes the maximum likelihood function a powerful tool.

The ability of the maximum likelihood function to identify the best location on a map directly relates to the performance of the GGI, the quality of the maps, and how much the gravity gradients vary among locations. The following figures offer one way to illustrate the phenomenon that makes this method possible. As the aircraft flies along its

true flightpath, the GGI measures and records the gravity gradients at discrete moments in time. Plotted out over time, the five independent components of the gravity gradient tensor might look like the signal shown in the top right corner of Figure 21. This particular signal comes from the computer program created to support this research effort, where the aircraft is flying at 150 meters per second and 5 kilometers over rough terrain with 1,800 meters per hour of INS drift and no GGI noise. This signal is unique to the aircraft's true flightpath and sensor, thus providing an opportunity to identify the aircraft's position on a map with an associated uncertainty.

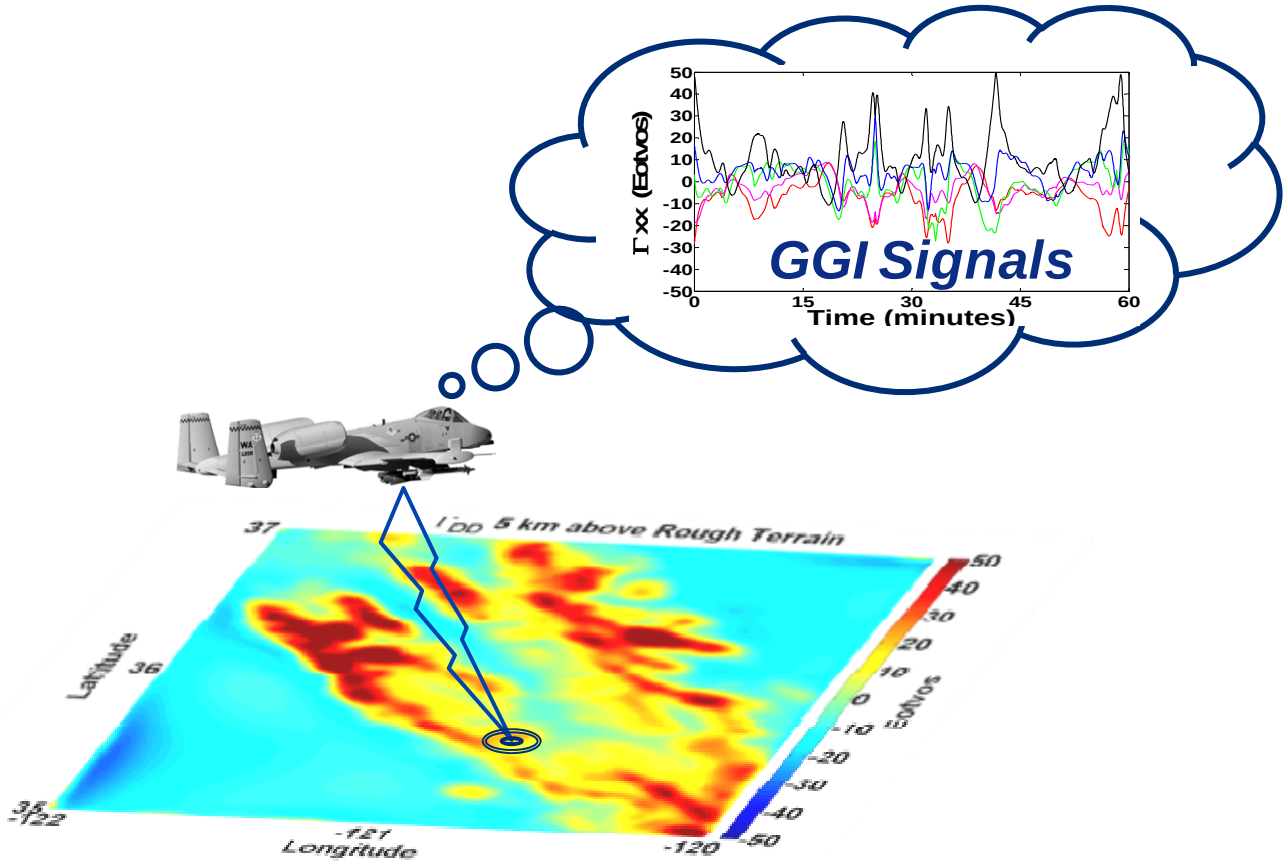


Figure 21: Matching a GGI Signal to a Map

A closer look at the GGI signal shows that the gravity gradients that would be encountered along the true flightpath differ from those encountered along a flightpath based on the position solutions of a drifting INS. In other words, if your navigation computer drifts far enough off course, relative to the noise levels of your GGI and maps, the map matching algorithm should be able to find a position solution where the measured and expected gravity gradients make a better match. Figure 22 shows the difference in GGI signals along an aircraft's true flightpath and INS flightpath as the INS approaches 1,800 meters of position error.

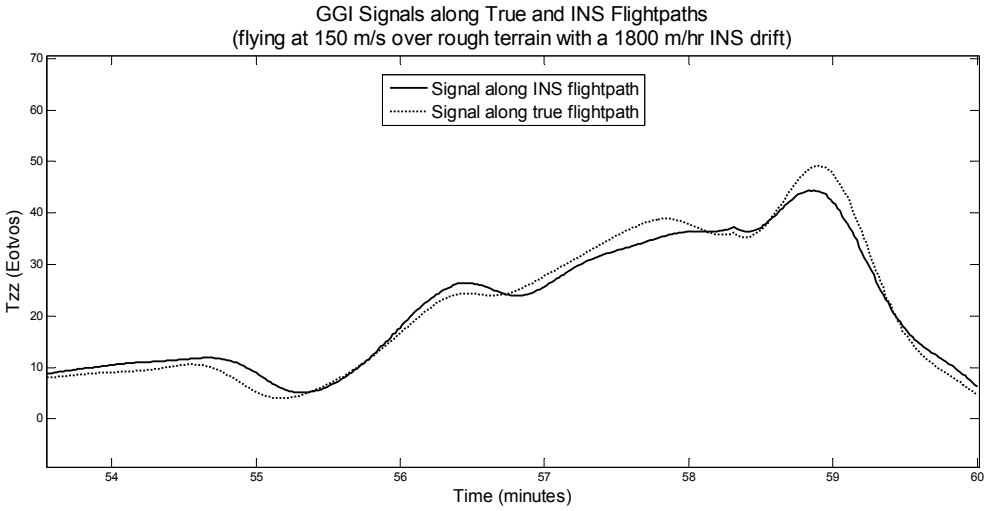


Figure 22: GGI Signals along True and INS Flightpaths

The map matching algorithm takes advantage of these unique signals by comparing GGI sensor data to gravity gradient maps. First, the algorithm loads a rectangular area of the truth maps in the database, based on the INS position solution and uncertainty. The algorithm selects data points from the truth maps based on the simulated resolution requested by the user. This allows the user to investigate the effects

of using maps with resolutions equal to or less than the truth maps. Then, the algorithm adds noise to simulate inaccuracies in the map based on the user's inputs. These map inaccuracies might result from imperfect models or actual measurements. Additionally, this approach assumes the position error of the data points on the gravity gradient maps are significantly less than the resolution of the map and within the simulated noise levels. Next, the algorithm interpolates the map to a resolution specified by the user. While an ideal interpolation would use an infinite resolution, constraints imposed by the likelihood function and computer processing power demand interpolation to finite resolutions. This interpolation allows the map matching algorithm to consider position solutions at higher resolutions than the maps provide. In other words, this allows the algorithm to arrive at position solutions between the posts available in the database of maps. The computer program uses a unique variable to communicate different map resolutions with the user. The Resolution Level corresponds to a specific map resolution as shown in Table 4.

Table 4: Measurements of Map Resolution

Resolution (Level)	Resolution (arcseconds)	North/South Post Spacing (~meters)	East/West Post Spacing (~meters)
7	0.046875	1	1
6	0.09375	3	2
5	0.1875	6	5
4	0.375	12	9
3	0.75	23	19
2	1.5	46	38
1	3	93	75
0	6	185	150
-1	12	370	300
-2	24	740	600
-3	48	1,480	1,200
-4	96	2,960	2,400
-5	192	5,920	4,800
-6	384	11,840	9,600
-7	768	23,680	19,200

The algorithm always selects an area of the truth maps such that at least 9 by 9 pixels are included in the likelihood function calculations. This prevents MATLAB from crashing on the interpolation commands. The minimum area ensures a statistically significant number of data points. The likelihood function calculates the likelihood for each pixel, and the algorithm selects the location with the maximum likelihood as the GGI solution.

Finally, the map matching algorithm calculates the uncertainty associated with the GGI position solution. The algorithm uses the posterior probability density function on page 44 and computer code modified from Storms' work[31]. The algorithm looks at the likelihoods for a line of pixels in the north and east directions, intersecting at the GGI position solution. This approach recognizes different uncertainties in different directions, which arise due to the aircraft's flightpath relative to map features. For example, if an aircraft flew over a ridgeline, the algorithm would have good information for determining position in a direction perpendicular to the ridgeline, but bad information for positioning parallel to the ridgeline. Consequently, uncertainty would be low in the perpendicular direction and high in the parallel direction. The algorithm forces a minimum uncertainty value equal to the resolution of the interpolated gravity gradient maps.

Kalman Filter

The computer program uses a discrete linear Kalman filter to determine the GAME position solution and uncertainty from the INS and GGI position solutions and uncertainties. The equations come from Grewal's text and are found on page 17. If the map matching algorithm fails to provide a unique position solution, the INS position solution and uncertainty become the GAME position solution.

Performance Measures

This research uses several measures to quantify GAME's performance. Some performance measures are also used with the GGI position errors to give an awareness of how gravity gradiometry and map matching perform alone. The computer program records position errors for the duration of flight and then reports the mean and standard deviation of the RMS position errors for the GGI and GAME solutions, as well as the 50th percentile CEPs (i.e. the median of the RMS position errors).

The computer program also introduces two new performance measures. The Performance Gain divides the INS CEP, as if it had drifted for the duration of flight, by GAME CEP. This effectively communicates how many times more accurate GAME's position solution is on average than an INS that worked for a length of time equal to the duration of flight. The calculation includes data from the entire duration of flight, because the performance gain aims to capture all effects, including effects before GAME reaches a steady state accuracy.

$$Performance\ Gain = \frac{INS\ CEP}{GAME\ CEP}$$

Equation 26: Performance Gain

The Break Even Point divides GAME's CEP by the INS drift rate. This effectively communicates how much time would pass before GAME's performance would start to be better than the INS working alone.

$$BEP = \frac{GAME\ CEP}{INS\ Drift\ Rate}$$

Equation 27: Break Even Point

Variables

To minimize confusion, this section provides more information about the variables discussed in this paper. The *terrain* variable selects whether the aircraft flies over the rough or smooth terrain discussed on page 52. The simulation flies the aircraft at a constant *velocity* and *altitude* measured above the average terrain height. *Flight duration* refers to how much time the aircraft flies in the simulation. *INS drift rate* sets how fast the INS position solution drifts away from the true position, as well as propagation of the INS uncertainty. The *position update rate* refers to how frequently the map matching algorithm runs, which also determines how frequently GGI solutions feed to the Kalman filter.

The *GGI components* variable lists or counts the number of components of the gravity gradient tensor that the map matching algorithm uses to calculate GGI solutions. *GGI noise* simulates the noise measured by the GGI onboard the aircraft. *Map noise* simulates the noise inherent in the maps carried in the aircraft's database. The values of the GGI and map noise variables reflect 1 standard deviation of white, Gaussian noise with zero mean. *Map resolution* refers to the simulated resolution of the gravity gradient maps stored in the aircraft's database. If the maps provided to the computer program have higher resolutions, the map matching algorithm will only use the resolution of information specified by this variable. The map resolution cannot be greater than the stored maps. The map matching algorithm then interpolates the gravity gradient maps until it achieves the resolution indicated by the *map interpolation* variable.

Sensitivity Analysis

A sensitivity analysis shows how each variable influences the outcome. In this research effort, the variables include terrain, altitude, velocity, flight duration, INS drift rate, position update rate, GGI components, GGI noise, map noise, map resolution, and map interpolation. The outcome refers to the GGI and GAME performance measures, in addition to other significant observations.

To perform this sensitivity analysis, all the variables will stay the same while one variable changes. In some situations, more than one variable will change at the same time to accommodate special circumstances and further understanding. In general, the variables that stay the same will be set to their default values. The left column of Table 5 lists the default values. The right column lists the values that will be included in each variable's sensitivity analysis.

Table 5: Sensitivity Analysis Variables

Default Value	Variable	Sensitivity Analysis Values
Rough	Terrain	Rough, Smooth
5	Altitude (km)	1, 5, 10, 15, 20, 25, 30
150	Velocity (m/s)	25, 50, 100, 150, 200…1250
2.2222	Flight Duration (hr)	0.25, 0.5, 1, 2, 4, 8, 16, 24, 32
2000	INS Drift Rate (m/hr)	0.2, 2, 20, 200, 2000, 20000
1	Position Update Rate	1, 15, 30 s, 1, 15, 30 min, 1 hr
5	GGI Components	Γ_{EE}, Γ_{NE}, Γ_{ED}, Γ_{ND}, Γ_{DD}, 2, 3, 4, 5
0.1	GGI Noise (Eö)	1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1, 5
0.01	Map Noise (Eö)	1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 0.5
3	Map Resolution (arcseconds)	3, 6, 12, 24, 48, 96, 192
3	Map Interpolation (arcseconds)	3, 1.5, 0.75, 0.375, 0.1875

The defaults reflect typical values. Rogers' optimistic prediction of the GGI performance expected to be available within the next decade provides the default value for GGI noise.[19] Map noise is one order of magnitude lower, under the assumption that surveys use more accurate sensors, collection methods, and post processing, compared to sensors onboard an aircraft that process data in real time. Regardless of the defaults, the sensitivity analysis provides insight into how the variables affect GAME performance.

Practical Simulations

In addition to the sensitivity analysis, practical simulations help in understanding GAME's performance in scenarios relevant to the Air Force. The first three include a fighter, cargo, and intelligence, surveillance, and reconnaissance (ISR) scenario. All three scenarios use typical values for the variables, as listed in Table 6. The fighter scenario varies INS drift rate and noise, cargo varies flight duration and noise, and ISR varies altitude and noise. Map noise remains one order of magnitude below the GGI.

Table 6: Practical Simulation Variables

Variable	Fighter	Cargo	ISR
Terrain	Smooth	Rough	Rough
Altitude (km)	5	10	5, 15, 25
Velocity (m/s)	400	250	150
Flight Duration (hr)	1.5	2, 4, 8, 16	24
INS Drift Rate (m/hr)	20, 200, 2000	2000	200
Position Update Rate	1	1	1
GGI Components	3	5	5
GGI Noise (Eö)	0.01, 0.1, 1	0.01, 0.1, 1	0.01, 0.1, 1
Map Noise (Eö)	0.001, 0.01, 0.1	0.001, 0.01, 0.1	0.001, 0.01, 0.1
Map Resolution (arcseconds)	3	3	3
Map Interpolation (arcseconds)	0.75	3	3

The practical simulations also include an optimistic and pessimistic look at the future. The values used in these scenarios will be based on the results of the sensitivity analysis, the technologies available today, and the technologies expected in the future.

Summary of Assumptions

The information in this section summarizes the assumptions discussed in previous sections and applicable to the modeling and simulation conducted in this research effort. The modeled gravity gradient maps represent realistic truth data, which was derived under the assumptions that DTED Level 1 adequately represents terrain effects, Earth's terrain is a constant density, gravity is a conservative field, and air's density is much smaller than Earth's. The map noise simulates inaccuracies in the map database, which might arise from imperfect modeling or surveying. Position errors in the map data points are small compared to the map's resolution and within the noise levels.

The GGI sensor provides information in the exact reference frame and accurately time stamped, or the errors are compensated and within the simulated noise levels. Changes in the simulation's true gravity gradients between the time the maps were created and the simulated flight are within the simulated noise levels.

The aircraft flies at constant velocity and altitude, and no INS drift occurs in the vertical direction. The INS position solution always drifts southeast at the rate specified by the user. Finally, aircraft dynamics do not affect the GGI measurements, or the effects are compensated and within the simulated noise levels.

IV. Results and Analysis

Appendix B presents a full table of the results. Since the methodology includes 10 variables with millions of permutations, this section limits discussion to information from the sensitivity analysis and practical scenarios. Despite this limitation, the scope provides a fundamental understanding of GAME's potential as an aircraft navigation aid.

This analysis scrutinizes all performance measures, but generally discusses results in terms of performance gain, since it provides a good basis for comparisons of overall performance. Since performance gains normalize GAME accuracy by the accuracy of an INS flying unaided for the duration of flight, values greater than 1 represent performance improvements. However, a theoretical minimum of 2 occurs in this analysis, because the computer program calculates GAME CEP using position errors from the entire duration of flight, while the INS CEP reflects position errors at the end of the flight. For example, if an INS drifted a constant 2 km/hr for 1 hour and the GGI solutions carried no weight, the INS and GAME CEPs would be 2 and 1, respectively. This results in a performance gain of 2, even though the GGI solutions did not improve upon the INS's performance.

By its definition, the performance gain makes a useful tool for deciding if GAME has good potential as an investment. If a scenario predicts a performance gain of 5-50, the investor must decide whether the investment in GAME for a 5 to 50-times accuracy improvement is worthwhile. If the same investment improves INS accuracy 3 times, then a performance gain of 5-50 might be a good investment. Performance gains less than 5 suggest that a comparable investment in other technologies might provide better returns. Although this paper does not estimate costs associated with improvements, Table 7 uses this logic to define three investment categories based on performance gain.

Table 7: Investing in Performance Gains

Potential Returns on Investments	*Performance Gain*
Excellent	> 50
Good	5 - 50
Poor	< 5

Terrain Effects

At default conditions, rough terrain provides a performance gain of 34.7, breaking even with the INS after 3.8 minutes. Smooth terrain provides a performance gain of 10.5, breaking even with the INS after 12.7 minutes. Both results suggest potential for good returns on investments, but the smooth terrain borders on poor. The GGI solutions offer accuracies with a CEP of 141 meters over rough terrain and 378 meters over smooth. From these perspectives, GAME appears to perform about 3 times better with rough terrain than smooth. This is great news for aircraft flying over rough terrain or long distances, because dynamic map features provide excellent information for accurate GGI solutions with low uncertainties. Unfortunately, the smooth terrain results provoke questions about worse case scenarios, such as high noise levels, high altitudes, less than all 5 components of the gravity gradient tensor, and terrain or water with even smoother map features.

The sensitivity analysis also provides a basis for terrain comparisons at different altitudes and different components of the tensor. When considering the best components, GAME performs 2 to 3 times better with rough terrain than smooth, whether using 1, 2, 3, 4, or 5 components. Figure 23 shows rough terrain's advantage decreasing as altitude increases, but GAME still performs 2 to 5 times better with rough terrain.

Altitude Effects

Figure 23 illustrates the decreases in GAME and GGI performance experienced with increases in altitude. The rate of performance loss appears to decrease at higher altitudes, which supports findings by Richeson that an altitude exists, relative to the GGI sensors noise levels, where terrain effects might be neglected. The high frequency information provided by terrain features at low altitude significantly improves GAME performance, but rapidly attenuates with increases in altitude.

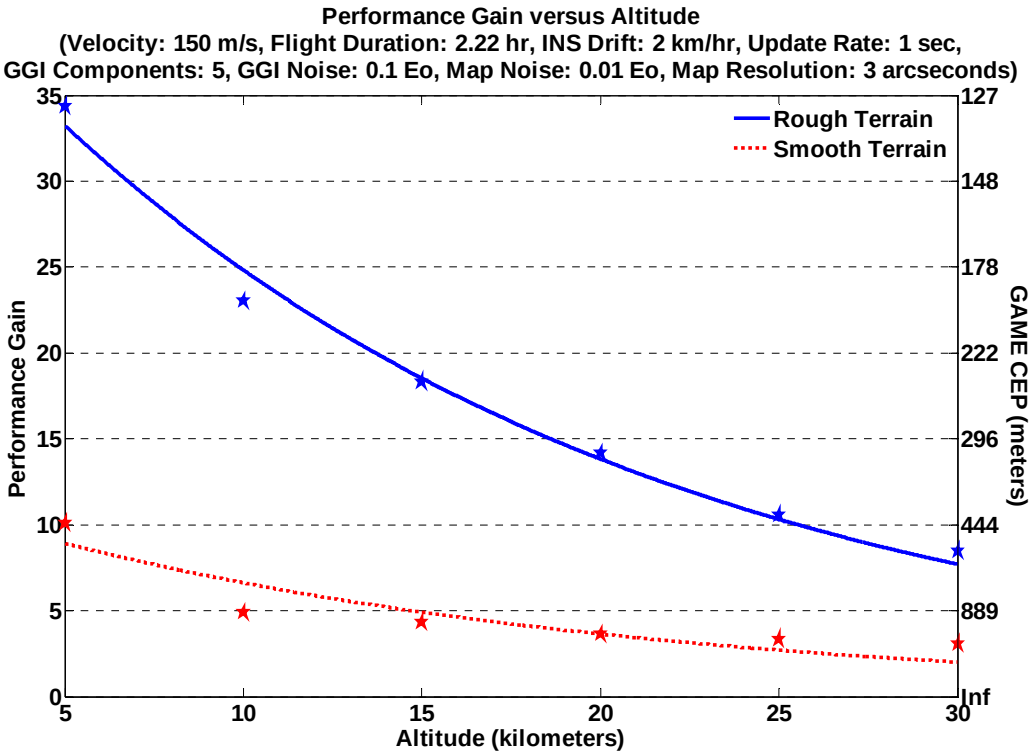


Figure 23: Effect of Altitude on Performance Gain

At 1000 meters, the map matching algorithm failed to find unique solutions at locations where terrain height exceeded altitude. This highlights a shortcoming in the simulation, since aviators generally do not fly through terrain. The successful GGI solutions at 1000 meters continue the trend of outperforming solutions at higher altitudes.

Velocity Effects

Velocity does not appear to affect GAME or GGI solutions, although noise drove small differences. The performance gain increases as velocity decreases, but not due to velocity. Instead, changes in flight duration, which ensure simulations cover the same terrain, mean an unaided INS would drift farther during the simulation. Thus, increases in performance gain reflect better returns on investments for longer flights.

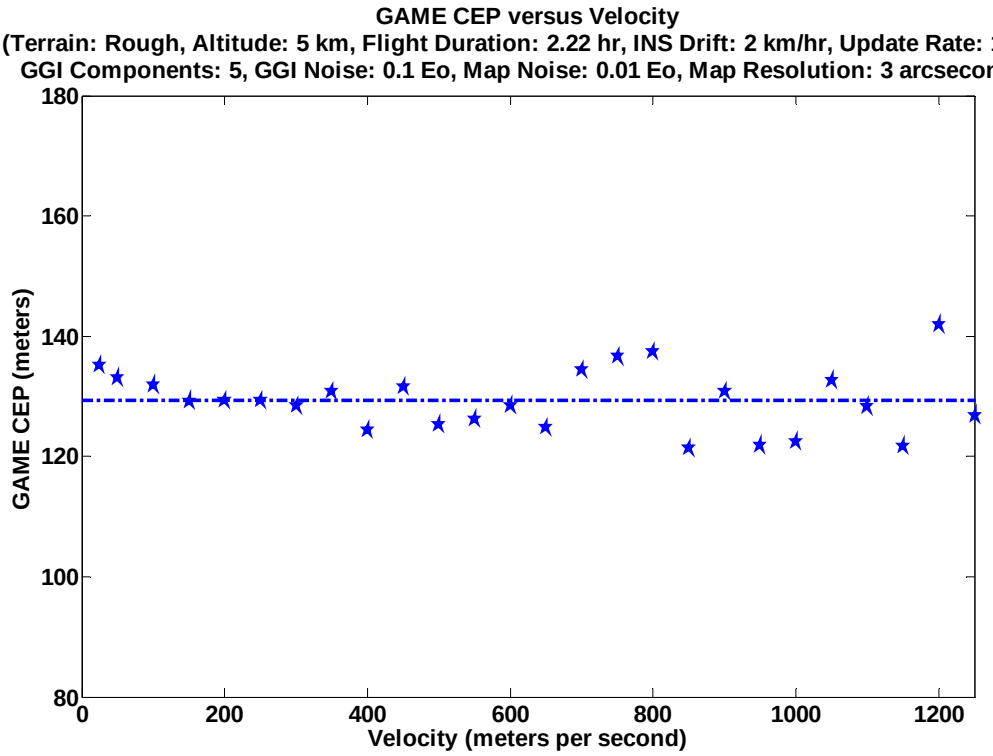


Figure 24: Effect of Velocity on GAME CEP

Although the results do not show that velocity affects accuracy, the simulation did not model the inner workings of a GGI. The methodology assumes accurate processing, recording, and time stamping of measurements. Noise might cover some of these errors, but the simulation maintained constant noise for all velocities. In reality, velocity might affect noise levels and introduce biases, which in turn affect GAME performance.

Flight Duration Effects

Flight duration only affected the performance gain. Even though the position solutions and uncertainties were not affected, longer flights with an unaided INS result in larger position inaccuracies. Thus, by definition of the metric, the performance gain increases with flight duration, because its accuracy grows relative to an unaided INS over longer periods of time. This increase in performance gain simply communicates that GAME provides greater potential returns on investments for longer flight durations compared to an unaided INS. At the default flight conditions, Figure 25 shows poor potential for returns on investments for flight durations less than about 30 minutes, good potential between 30 minutes and 4 hours, and excellent potential greater than 4 hours.

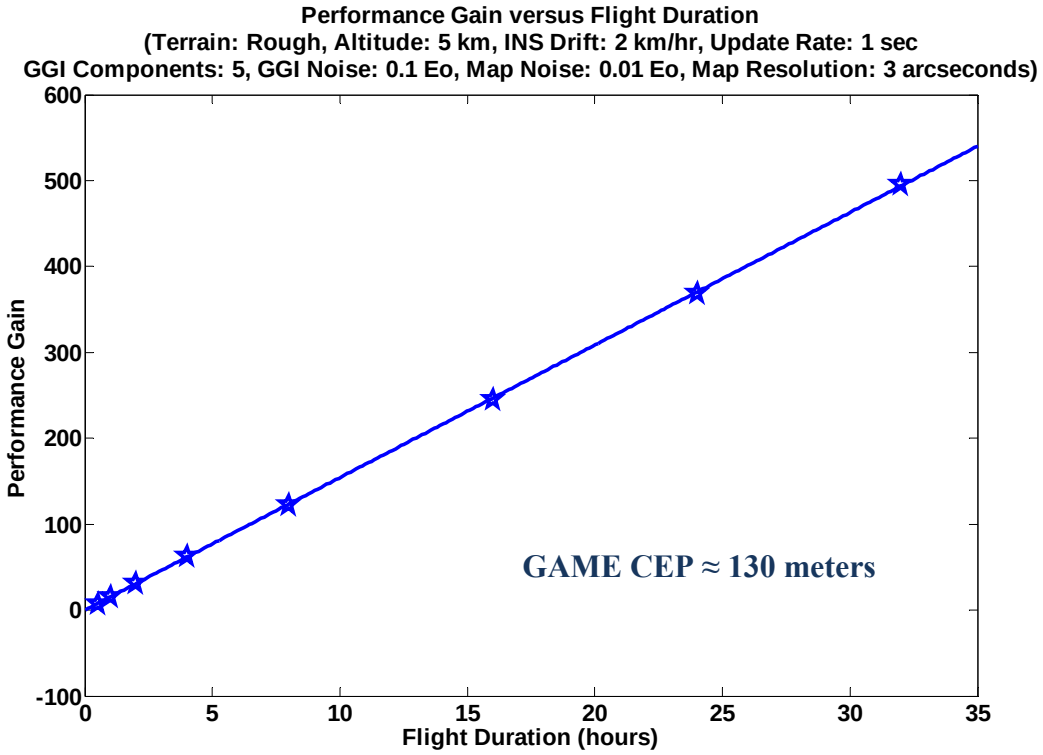


Figure 25: Effect of Flight Duration on Performance Gain

INS Drift Rate Effects

Similar to flight duration, larger INS drift rates increase performance gain, even though GGI solutions do not significantly change. This indicates that GAME provides greater returns on investments when working with a less accurate INS. Ironically, a more accurate INS improves GGI and GAME accuracy. At the default flight conditions, INS drift rates less than about 300 meters per hour result in poor returns on investments. At 200 meters per hour, it takes 37 minutes just for the GGI solutions to break even with an unaided INS. At 20 meters per hour and below, the performance gain bottoms out at the improvement threshold. The simulation at 20 meters per hour dips slightly below, indicating that GAME decreased accuracy. Hope is not lost for scenarios with a highly accurate INS, because changes to other variables promise higher performance gains, especially longer flight durations and higher map resolutions.

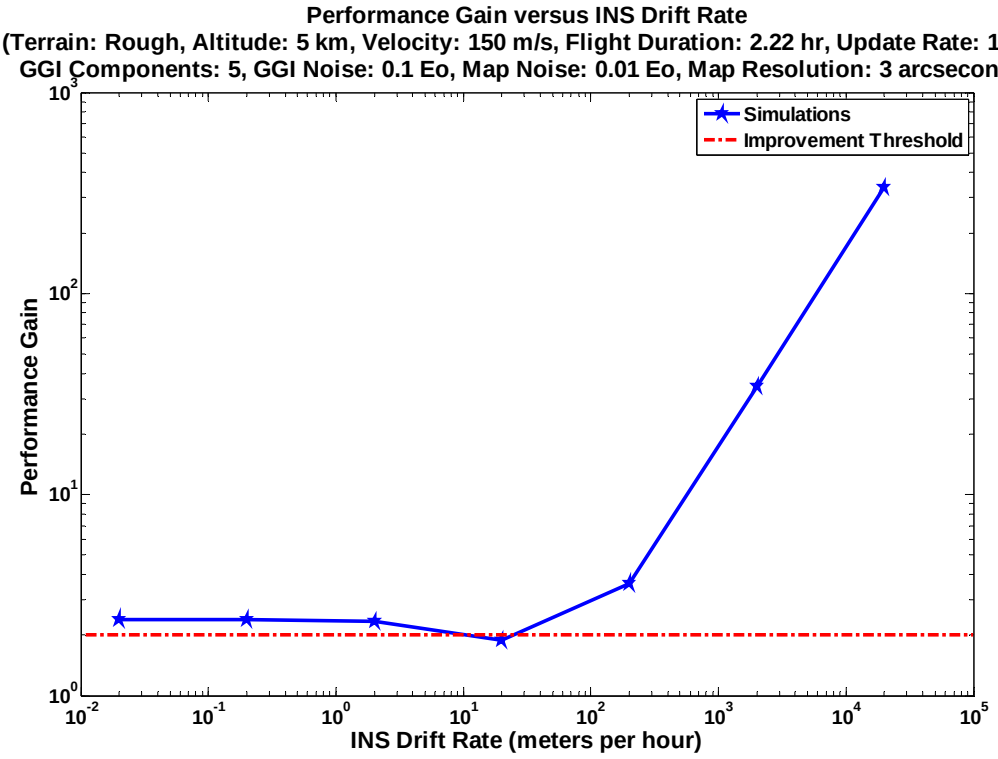


Figure 26: Effect of INS Drift Rate on Performance Gain

Position Update Rate Effects

More frequent updates improve all of the performance measures, except GGI solution performance, which remains statistically neutral. At the default flight conditions, only position updates every second offer potentially excellent returns on investments. Poor potential exists for position update rates less frequent than once every minute. This, of course, suggests that efficient algorithms and fast computer processors directly affect GAME performance. Although producing GGI solutions once every second took double the processing time of the other simulations, the update rate did not appear to affect computer processing times for updates rates less frequently than every 15 seconds. Updating the position less frequently decreases the number of times the algorithm runs, but increases the size of the map searched for a match.

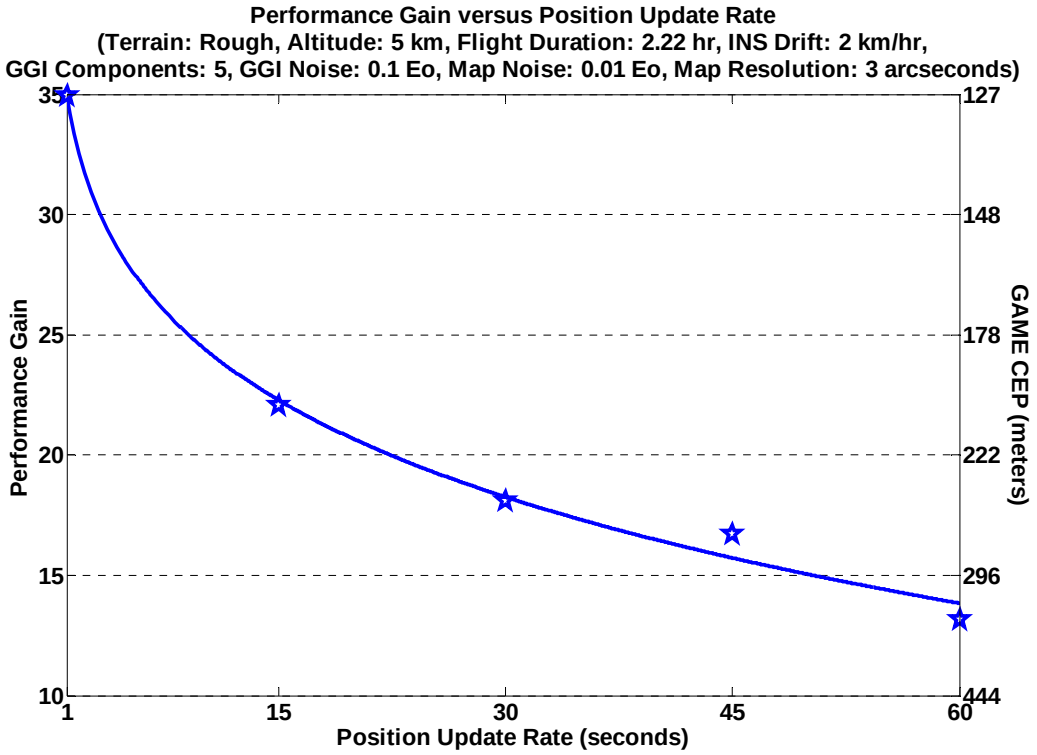


Figure 27: Effect of Position Update Rate on Performance Gain

GGI Component Effects

Looking at the best performing components, GGI and GAME performance generally increases with the number of components of the gravity gradient tensor included in the simulations. However, each increase in the number of components results in less increase. Even though using all five components produces the best results, using three components appears to offer the best value, under the assumption that each increase in the number of components comes at a proportional price.

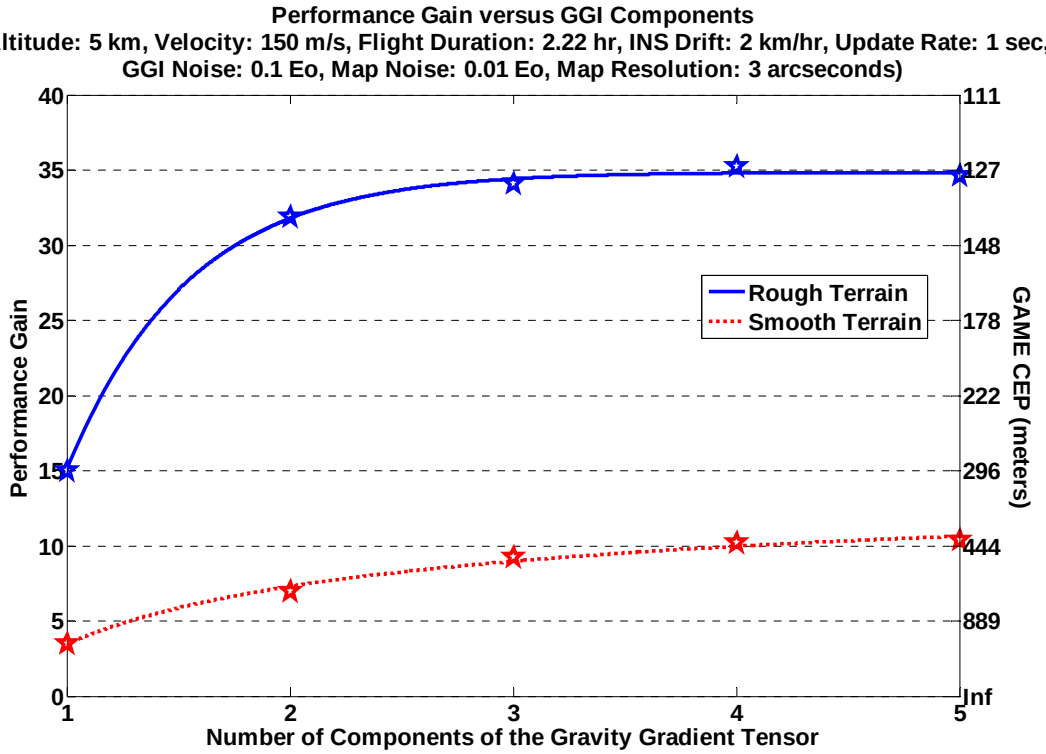


Figure 28: Effect of GGI Components on Performance Gain

This particular sensitivity analysis also provides an opportunity to analyze which components result in the best performance. In the north, east, down reference frame, the following tables list the components in order of their performance gains.

Table 8: Best GGI Components over Rough Terrain

	Γ_{DD}	Γ_{ND}	Γ_{NE}	Γ_{EE}	Γ_{ED}
Performance Gain (ND)	15.0	11.5	7.7	4.9	4.7
RMS Mean (meters)	683	818	1029	1063	994
RMS Std Dev (meters)	509	644	830	841	786
CEP_{50} (meters)	564	638	804	832	785

Table 9: Best GGI Components over Smooth Terrain

	Γ_{ND}	Γ_{DD}	Γ_{NE}	Γ_{ED}	Γ_{EE}
Performance Gain (ND)	3.52	3.46	2.99	2.98	2.65
RMS Mean (meters)	2003	1969	2636	2436	2688
RMS Std Dev (meters)	2061	1999	2469	2317	2452
CEP_{50} (meters)	1265	1274	1849	1694	1963

Γ_{DD} and Γ_{ND} take first and second place over rough and smooth terrain, respectively, while Γ_{EE} and Γ_{ED} take fourth and fifth. The relative importance of the components diminishes as the terrain smoothens, and the ranking order changes when ranking by RMS mean and CEP. This indicates that even though one component might result in more accurate position solutions, the associated uncertainties might be higher. From this perspective, standard deviations also play a role in how the rankings appear different when considering different performance metrics.

The individual ranks of the components do not necessarily correspond with which combinations of components work together the best, since different components might perform better in different locations. For example, if Γ_{DD} and Γ_{ED} performed well in different locations, they might make a stronger pair than Γ_{DD} and Γ_{ND} performing well only in the same locations. Thus, Appendix B includes combinations of 2, 3, and 4 components at the default conditions. The appendix includes all combinations for 3 and 4 components, but only combinations with Γ_{DD} for 2 components, since there are so many

combinations. Table 10 presents the best performing combinations, which do not always follow the logic of the individual rankings. For example, one might assume that Γ_{DD} and Γ_{ND} make the best duo over rough terrain, but Γ_{EE} takes the place of Γ_{ND}. In fact, second-ranked Γ_{ND} doesn't even make it into the best trio or quartet!

Table 10: Best Combinations of GGI Components

	Rough Terrain	Smooth Terrain
1 Component	$\boldsymbol{\Gamma_{DD}}$	$\boldsymbol{\Gamma_{ND}}$
2 Components	$\boldsymbol{\Gamma_{DD}}$, $\boldsymbol{\Gamma_{EE}}$	$\boldsymbol{\Gamma_{DD}}$, $\boldsymbol{\Gamma_{ED}}$
3 Components	$\boldsymbol{\Gamma_{DD}}$, $\boldsymbol{\Gamma_{EE}}$, $\boldsymbol{\Gamma_{NE}}$	$\boldsymbol{\Gamma_{DD}}$, $\boldsymbol{\Gamma_{ED}}$, $\boldsymbol{\Gamma_{ND}}$
4 Components	$\boldsymbol{\Gamma_{DD}}$, $\boldsymbol{\Gamma_{EE}}$, $\boldsymbol{\Gamma_{NE}}$, $\boldsymbol{\Gamma_{ED}}$	$\boldsymbol{\Gamma_{DD}}$, $\boldsymbol{\Gamma_{ED}}$, $\boldsymbol{\Gamma_{ND}}$, $\boldsymbol{\Gamma_{EE}}$

The rankings in these simulations do not necessarily hold for other scenarios. The 5 independent components of the gravity gradient tensor perform differently in different situations. The hypothetical prism on page 31 gives a good indication that map feature dynamics vary for components in different situations. While map features are a function of location, map quality and resolution also play a role in determining which components and combinations perform best in given situations. However, in all the simulations, Γ_{DD} makes it into the best performing trio and quartet. Richeson agrees that Γ_{DD} varies more than the other components, suggesting that it also performs better. However, he points out that the components appear to vary the most in the same locations, suggesting that the other components perform the best in the same locations as Γ_{DD}.[12]

GGI Noise Effects

At the default conditions, decreases in GGI noise improved all performance measures down to about 0.01 Eö. Beyond that point, decreases in GGI noise did not significantly improve results. From the perspective of the sensitivity analysis, this observation communicates that, beyond a certain point, decreases in GGI noise levels do not significantly improve results, unless other variables also improve (e.g. map noise levels, map resolutions, and map interpolation). In other words, despite improvements in GGI noise levels, weaker links in other areas might limit GAME performance.

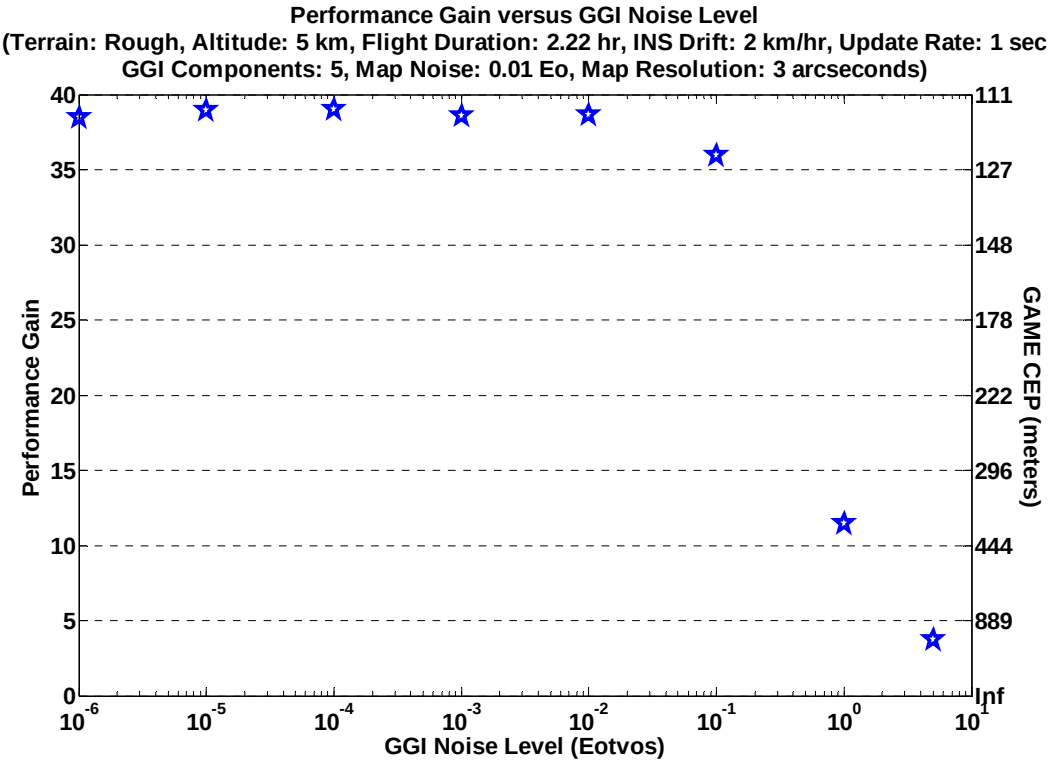


Figure 29: Effect of GGI Noise on Performance Gain

During this portion of the sensitivity analysis, the map matching algorithm crashed. Troubleshooting traced the source of the crashes to the algorithm's failure to

identify a unique position solution, which was caused by the likelihood function essentially rejecting all position solution candidates as a possible match. This occurred because the likelihood function only included the GGI noise levels in its calculations. Thus, when the differences between the measured and expected (i.e. the sensor and map) values were much higher than GGI noise levels, all position solution candidates were rejected. To better tune the map matching algorithm, the likelihood function was modified to include the sum of the GGI and map noise levels. This is a practical modification, assuming the approximate noise levels of the GGI and map are known.

$$L(y_t / x_t) = \frac{1}{\sqrt{(2\pi)^N (\sigma_{GGI} + \sigma_{map})^2}} e^{-\frac{1}{2(\sigma_{GGI}+\sigma_{map})^2}\sum_{k=1}^{N}[y_{t,k} - h_k(x_t)]^2}$$

Equation 28: Likelihood Function as Applied in the Algorithm

The modification significantly decreases the number of failed map matches and enables successful simulations at lower noise levels. After the modification to the map matching algorithm, all simulations were rerun, so the results presented in this paper all use the same algorithm. The new GGI position solution and uncertainty results did not appear to significantly change compared to the results before the algorithm's modification, except that more successful map matches occurred, resulting in more position updates to the Kalman filter and better GAME performance at low noise levels.

Overall, the GGI noise sensitivity analysis shows how sensor performance affects GAME solutions. Under the assumptions of this research, this includes uncompensated

effects of aircraft dynamics, but not mass movements onboard the aircraft. Since masses onboard the aircraft would be relatively close to the GGI, even small movements could significantly affect sensor measurements. While small distances between differenced accelerometers and other techniques minimize the effects of aircraft dynamics, Figure 30 illustrates what attention to detail is required to compensate for mass movements. The figure applies the derivative of Newton's Law of Gravitation in the same manner as Richeson and plots selected masses over a range of distances.

$$|\Gamma| \approx \frac{2GM}{l^3}$$

Equation 29: Gravity Gradient Approximation

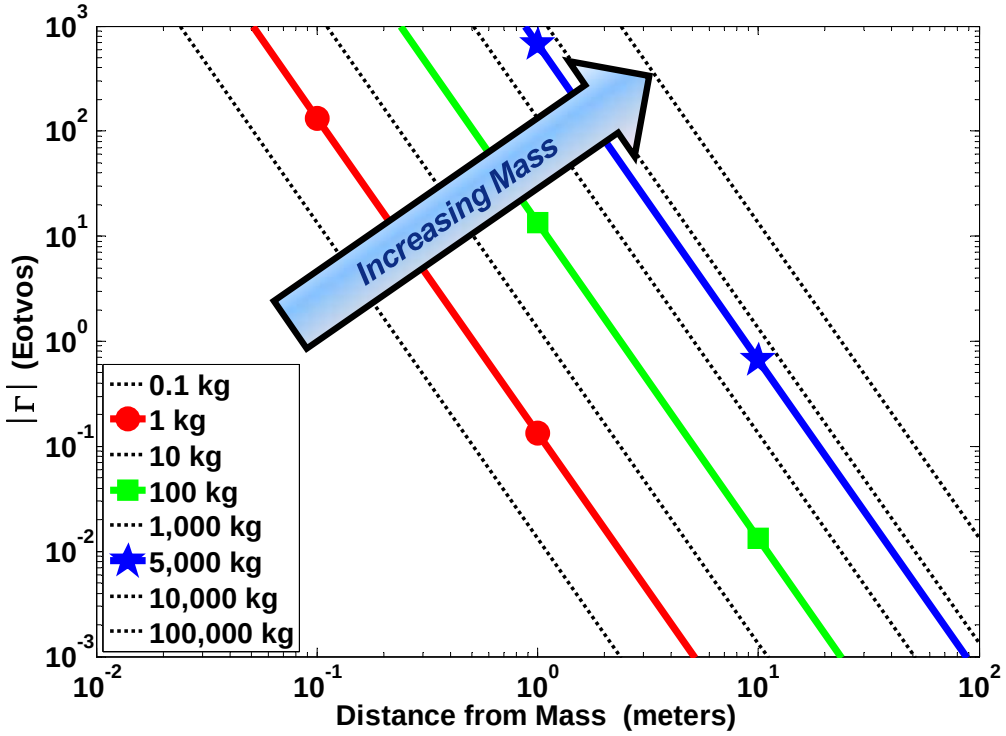


Figure 30: Mass Movements Onboard an Aircraft

Assuming a 0.1 Eö noise level already exists in the GGI, the effects of 0.1 and 1-kilogram masses 2 meters or more away from the sensor would be at or below the GGI's noise levels. Depending on how much they move, 100-kilogram masses significantly affect gravity gradients within about 5 meters. Phenomena in this category might include landing gear retraction, movement of personnel, or employment of small munitions. Assuming an aircraft uses 10,000 pounds of fuel (i.e. 4,536 kilograms) or more during a mission, the effects on gravity gradients almost always soar above the noise levels. Other mass movements to consider include shifting cargo, flight controls, propulsion systems, and flying in close formation. Options to compensate for mass movements onboard or in close proximity to an aircraft might include feeding mass movement information to the computer, placing the GGI in a location far away from moving masses, and improving the map matching algorithm to deal with static and transient biases. In general, the aircraft could act as a bias and calibration of the sensor onboard the aircraft might be required. Other methods of calibration include computing expected gravity gradients at a known location or comparing sensor outputs to a surveyed location before flight.

Map Noise Effects

Similar to the effects of GGI noise, decreases in map noise improve performance measures down to about 0.01 Eö. Beyond that point, decreases do not significantly improve results. This observation communicates that, beyond a certain point, decreases in map noise levels do not significantly improve results, unless other variables also improve (e.g. GGI noise levels, map resolutions, and map interpolation). In other words, weaker links in other areas might drive inaccuracies, despite improvements in map noise.

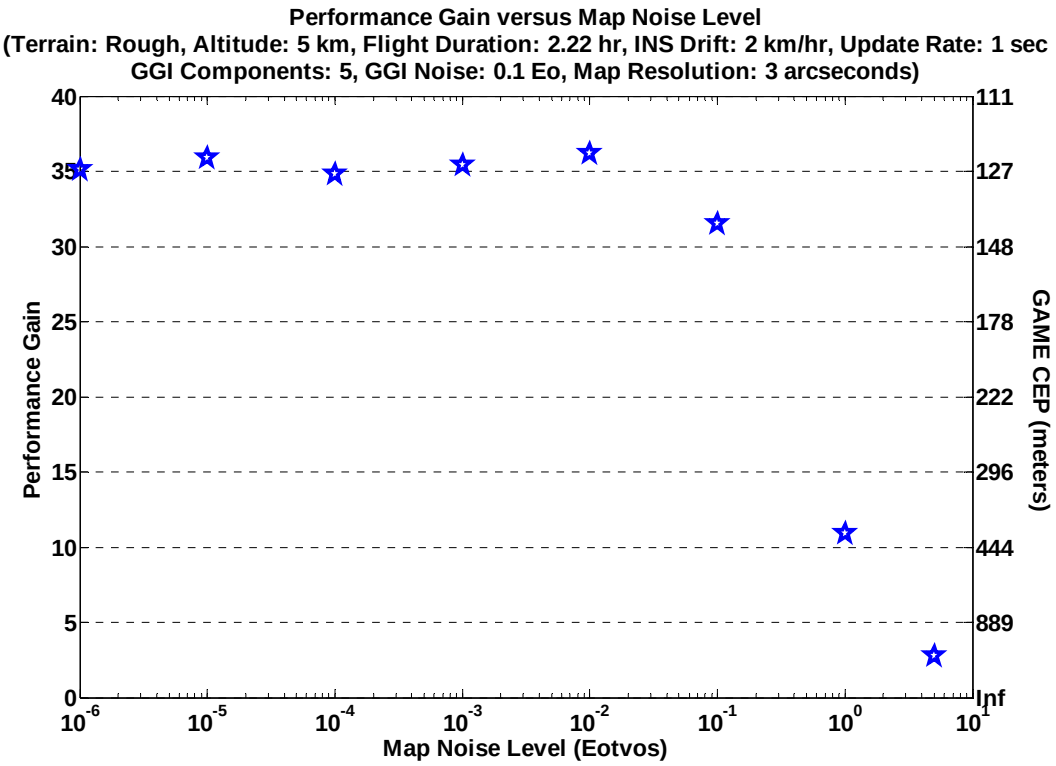


Figure 31: Effect of Map Noise on Performance Gain

Under this paper's assumptions, map noise represents the cumulative effects of inaccurately modeled maps or measured gravity gradients (e.g. a noisy GGI used in map-making surveys), inaccurately positioned data points, and gravity gradient changes from the time of the map's creation to GAME employment. The latter error source raises the question, "How much mass movement does it take to affect gravity gradients?" In general, mass movements might be manmade, geological, or astrological. Examples include new construction projects, especially sky scrapers and dams, quarries and landfills, and the movement of massive ships, aircrafts, and satellites; continental drifts, volcanic eruptions, melting glaciers, and ocean tides; the sun and moon. The following figure, based on Equation 29, provides some insight into what masses at what distances might significantly change gravity gradients, depending on how much they move.

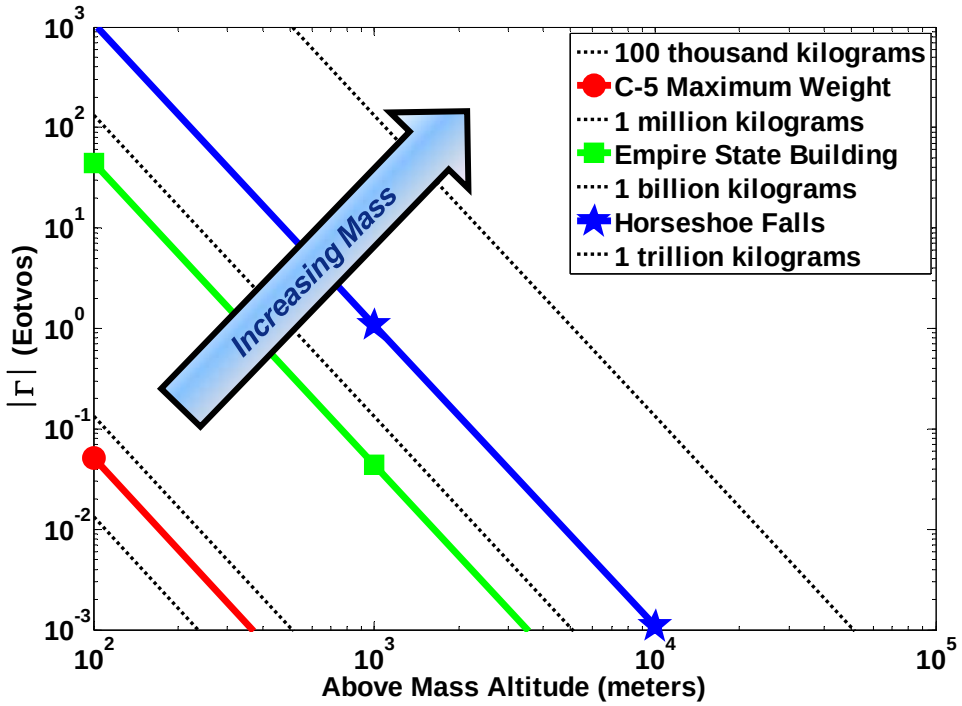


Figure 32: Mass Movements affecting Gravity Gradient Maps

C-5 Maximum Weight, Empire State Building, and Horseshoe Falls represent 420 tons (381 thousand kilograms), 365 thousand tons (331 million kilograms), and 9 million tons (8.2 billion kilograms), respectively. If a 0.01 Eö of noise already exists in the maps, the movement of a large cargo aircraft, like the C-5, would be at or below noise levels at a distance of 200 meters or more. If GAME flies within 2 kilometers of a new structure the mass of the Empire State Building, additional map inaccuracies should be expected above the noise levels. A major geological event that moves 9 million tons of mass, such as the mass of water that flows over Horseshoe Falls in 1 hour, causes map changes above noise levels for aircraft flying below 6 kilometers, depending on how far the 9 million tons moves. Robust map matching algorithms and map corrections for large mass movements could minimize the effects of mass movements on GAME performance.

Map Resolution Effects

While holding the interpolated map resolution constant, the simulated map resolution does not appear to affect performance measures until the simulated resolution decreases to Resolution Level 3. This represents a resolution with post spacing greater than approximately 1,200 meters. Since the interpolations are linear, this suggests that an insignificant amount of information is lost when using gravity gradient maps with resolutions as low as 48 arcseconds, compared to 3-arcsecond maps. While this attests to the effectiveness of map interpolation, it cannot demonstrate how much information would be gained with map resolutions higher than Resolution Level 1. Assuming information occurs at different frequencies, higher resolutions might provide more information and improve GAME performance beyond the results presented in this paper.

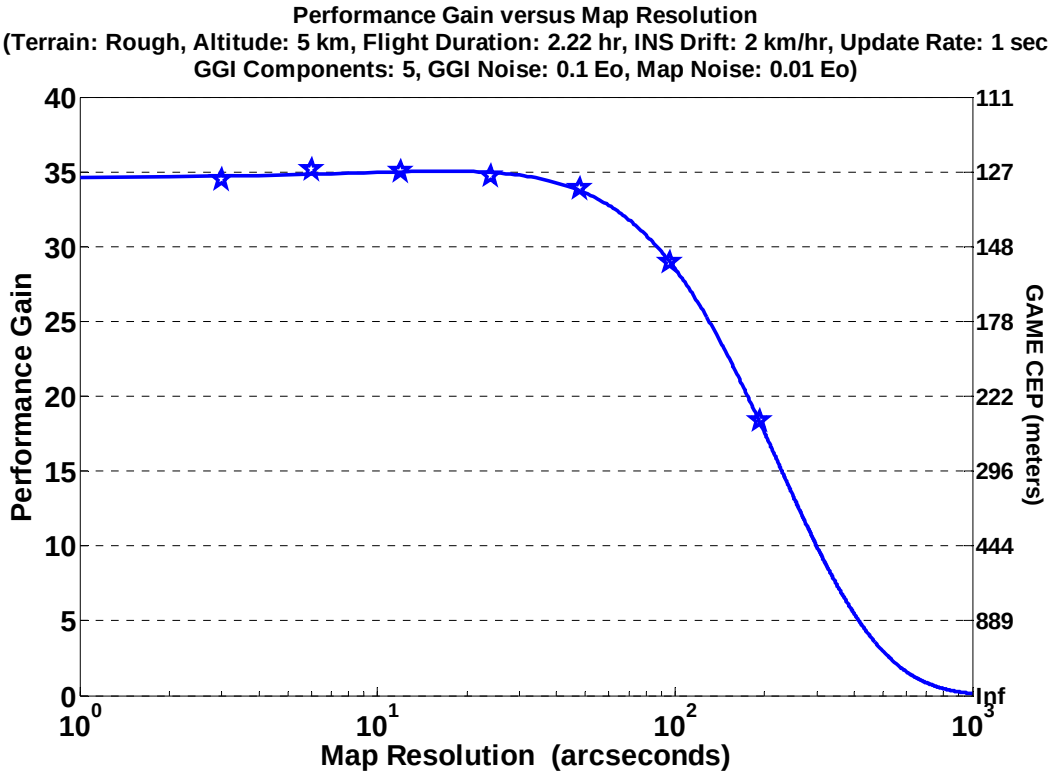


Figure 33: Effect of Map Resolution on Performance Gain

Map Interpolation Effects

Intuitively, allowing the map matching algorithm to interpolate the available maps to identify more accurate position solutions should improve performance. However, at the default flight conditions, no significant improvements occur. A closer look at the simulation's outputs reveals that noise levels approach and frequently exceed the differences in gravity gradients from point to point on the maps. Despite interpolation, this keeps the accuracy of GGI position solutions at the mercy of the system's random noise. Similar to the results seen in the GGI and map noise sections, improvements in map interpolation do not significantly improve results beyond a certain point, unless other variables also improve (i.e. GGI and map noise levels, map resolution, and the capabilities of the map matching algorithm).

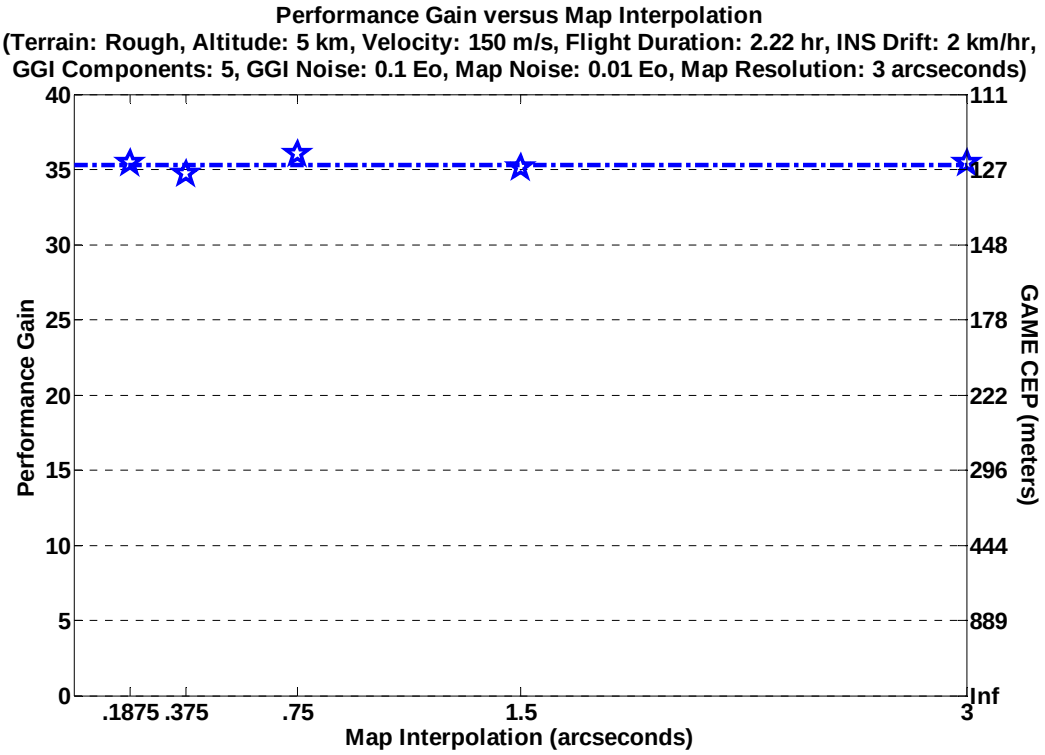


Figure 34: Effect of Map Interpolation on Performance Gain

Fighter Mission Performance

In the fighter scenario, simulations indicate that GAME provides poor potential for returns on investments for most combinations of GGI noise levels and INS drift rates. The scenario assumes that a typical fighter mission flies a 1.5-hour mission over smooth terrain at an altitude of 5 kilometers and velocity of 400 meters per second. The map matching algorithm interpolates 3-arcsecond maps to 0.75 arcseconds and only uses three components of the gravity gradient tensor (Γ_{DD}, Γ_{ED}, and Γ_{ND}).

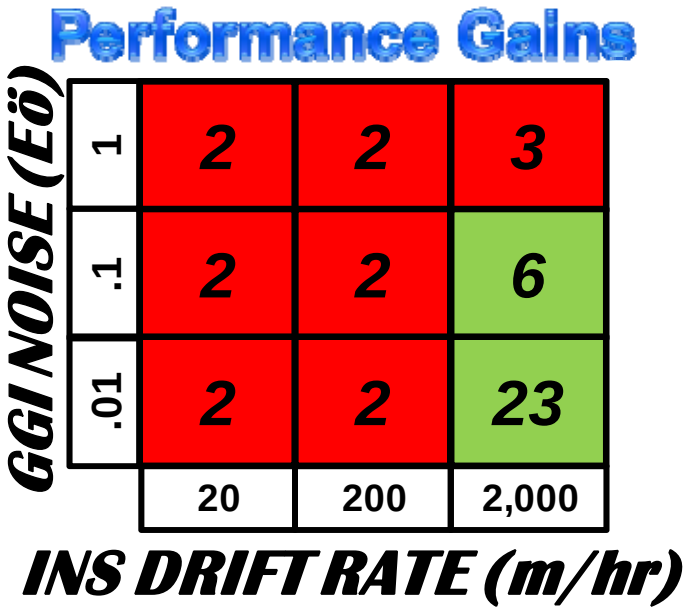


Figure 35: Performance Gains on a Fighter Mission

Figure 35 presents the performance gains at various GGI noise levels and INS drift rates. Results for a scenario involving 0.1 Eö and less of GGI noise and a 2,000 meter per hour INS drift rate suggest good potential returns on investments. However, this describes an unlikely scenario where the performance of airborne GGIs improves an order of magnitude over today's GGIs, while INS performance remains static at today's levels. The scenario involving 0.1 Eö of GGI noise and a 2,000 meter per hour INS drift

rate shows that GAME provides a 6-times improvement over the unaided INS. Figure 36 shows the GAME position errors versus time under these particular conditions, where an INS of today's caliber works with a GGI noise level expected to be available in the near future.

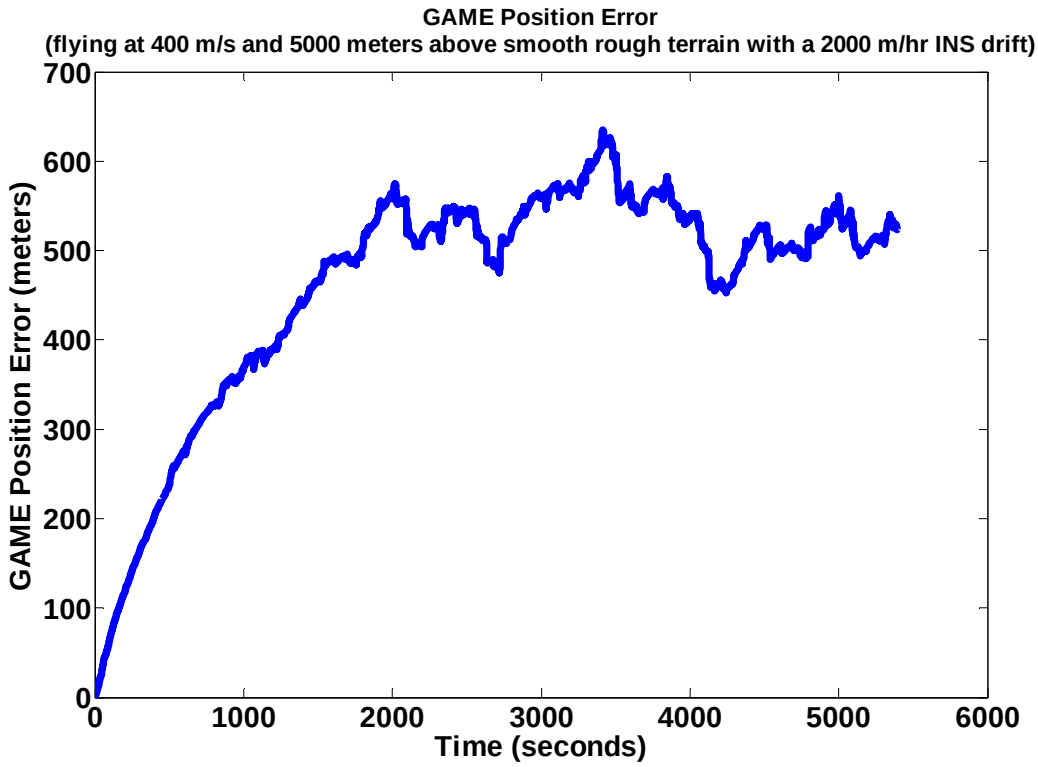


Figure 36: GAME Position Accuracy on a Fighter Mission

Figure 36 illustrates how GAME bounds the INS drift at about 510 meters. After a 1.5-hour sortie, the unaided INS would have drifted 3,000 meters. At the given conditions, GAME's steady state accuracy provides a respectable and enduring capability to fighter aircraft. In general, all of the simulations in the fighter scenario provide respectable GGI position accuracies, as seen in the following figure. This suggests that

fighter missions under other conditions, such as rougher terrain and longer flight durations, could still achieve high performance gains.

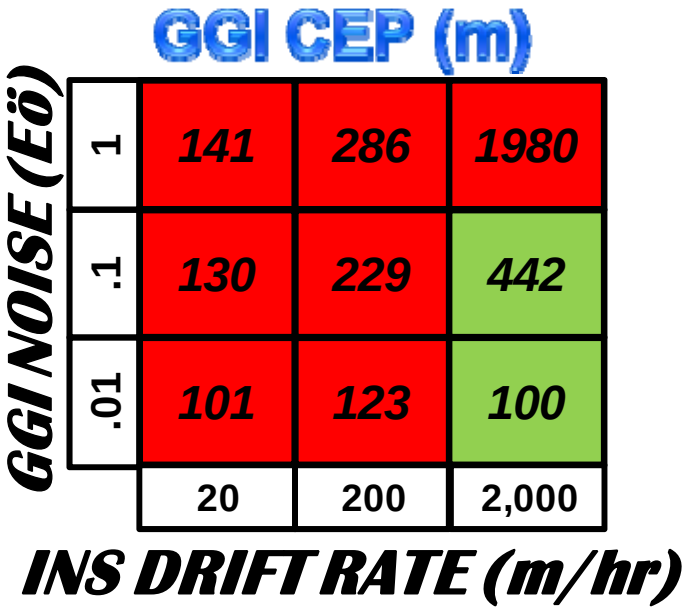


Figure 37: GGI Position Accuracies on a Fighter Mission

Cargo Mission Performance

In the cargo scenario, simulations indicate that GAME provides good to great potential for returns on investments for all combinations of GGI noise levels and INS drift rates. The scenario assumes that a typical cargo mission flies over rough terrain at an altitude of 10 kilometers, cruises at 250 meters per second, and carries an INS with a 2 kilometer per hour drift rate. The map matching algorithm uses all five components of the gravity gradient tensor and 3-arcsecond maps with no interpolation.

Figure 38 presents the performance gains at various GGI noise levels and flight durations. Only the results for a scenario involving 1 Eö of GGI noise and a 2-hour flight duration borders on a poor potential return on investments. However, this describes a

relatively short cargo mission with a GGI near the caliber of technologies available today. Flight durations 4 hours and longer with this caliber of GGI already promise good returns on investments. Considering GGI noise levels expected to be available in the near future, the simulations suggest excellent potential for returns on investments.

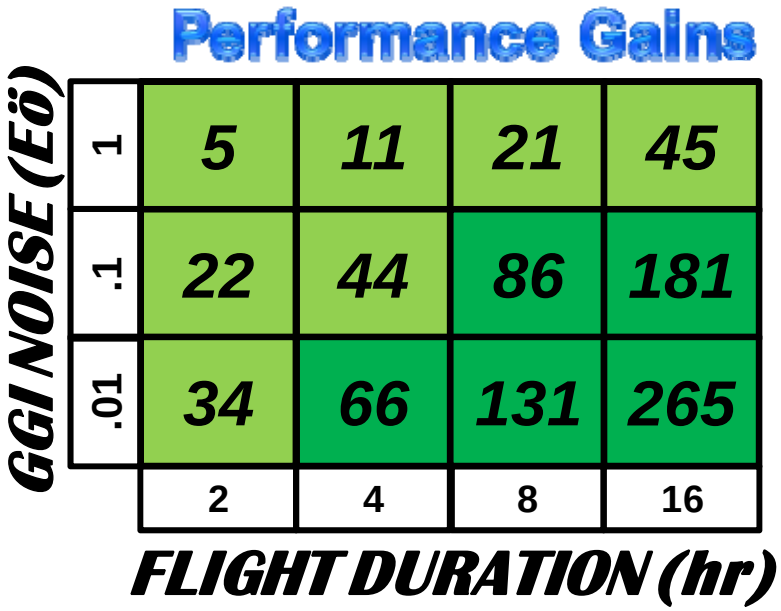


Figure 38: Performance Gains on a Cargo Mission

The prospect of GAME providing good returns on investments with today's technologies demands a closer look. A transcontinental flight might fall into the 8-hour flight duration category, which estimates a performance gain of 21. These conditions offer an associated break even point of 23 minutes, a GAME CEP of 754 meters, and a GGI CEP of 2.3 kilometers. These CEPs may sound high, but are within reach of today's technologies and provide a bounded error that's not bad for flying an equivalent distance of 4,500 miles. Even better, these CEPs represent a steady state and endure for as long as the aircraft flies. Figure 39 shows GAME position errors versus time for the cargo mission simulation with a 1 Eö GGI noise level and 8-hour flight duration.

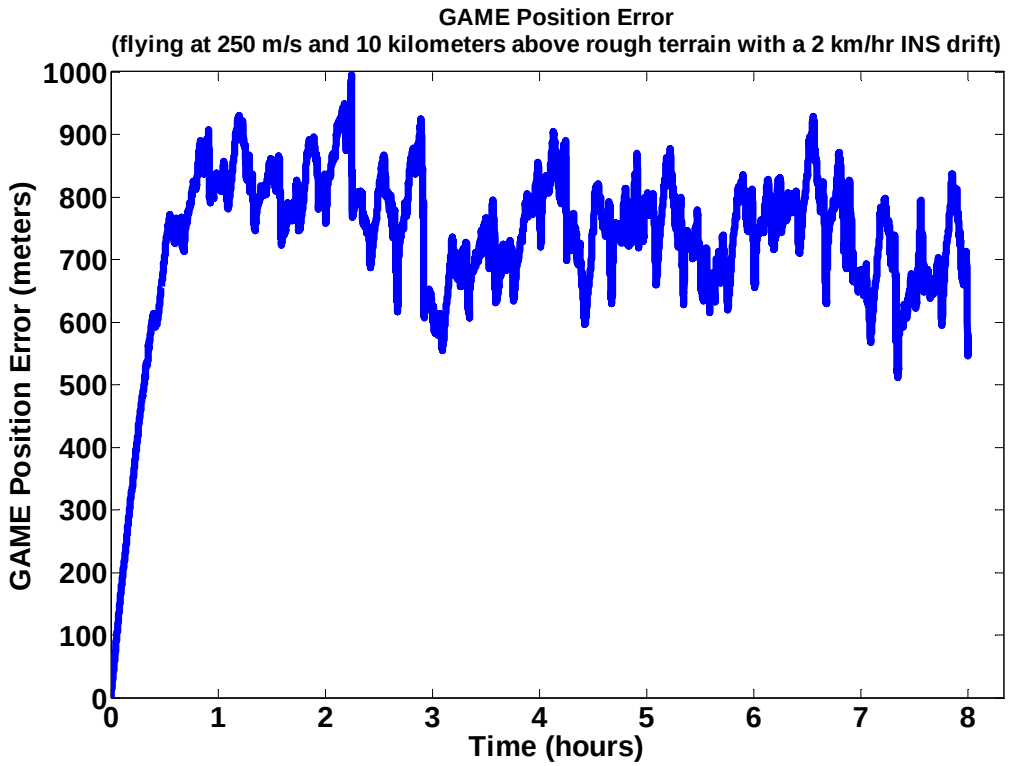


Figure 39: GAME Position Accuracy on a Cargo Mission

Of course, many long distance flights cross oceans, which raises questions regarding performance. The cargo simulations use rough terrain, because cargo aircraft probably encounter rough map features from time to time that result in large position updates, reminiscent of a saw-tooth curve. Richeson points out that gravity gradients offer map features over water,[12] while other map-based aids do not. In addition to the geoid's long wavelength gradients, the ocean floor contributes to map features. The National Oceanic and Atmospheric Administration estimates the average ocean depth at 4,267 meters and the deepest trench at 11,030 meters.[35] Three of Richeson's figures, beginning on page 33 in this paper, show the effects of underwater terrain on gravity gradients. Using Figure 23 from the sensitivity analysis on page 67, the effects of ocean depth on performance gain can be approximated by an equivalent increase in altitude.

ISR Mission Performance

In the ISR scenario, simulations show that GAME generally provides good potential for returns on investments. The scenario assumes that a long endurance unmanned aerial vehicle conducts intelligence, surveillance, and reconnaissance over a 24-hour flight duration. Similar to the reasoning for the cargo scenarios, the simulations use rough terrain under the assumption that long missions periodically encounter rough map features. The ISR simulations also fly at 15 km altitude, 150 meters per second, use all five components of the gravity gradient tensor and 3-arcsecond maps with no interpolation, and carry a cutting edge INS with only 200 meters per hour of drift.

Figure 40 presents the performance gains at various GGI noise levels and altitudes. All scenarios estimate good potential for returns on investments, except for when aircraft employ a GGI with a 1 Eö noise level at or above about 15 kilometers altitude. Tomorrow's GGIs appear to be a good investment for long endurance missions.

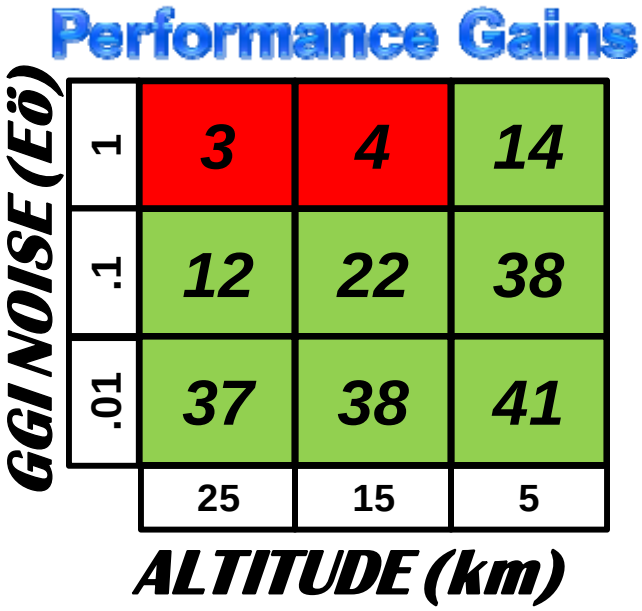


Figure 40: Performance Gains on a ISR Mission

The ISR simulations provide an opportunity to illustrate the effects of altitude on GAME solutions, as well as other variables that specifically decrease the accuracy of GGI solutions. With GGI noise levels of 0.1 Eö, Figure 41 shows how lower altitudes lead to more accurate GAME solutions at 5 kilometers altitude (CEP = 126 meters), compared to the results at 25 kilometers (CEP = 416 meters). GAME reaches steady state accuracy quicker at 5 kilometers altitude, versus 25 kilometers. In Figure 42, GGI solutions possess significantly more accuracy at 5 kilometers (CEP = 141 meters) than 25 kilometers (CEP = 698 meters). The GGI solutions also experience an initial ramping up, which relates to the INS's initially superior accuracy preventing the map matching algorithm from searching larger areas for less accurate potential solutions.

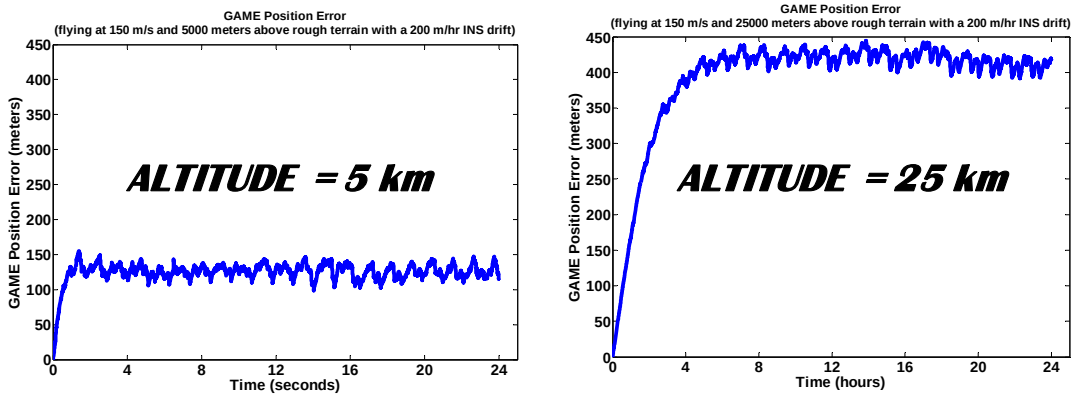


Figure 41: Altitude's Effect on GAME Solutions on an ISR Mission

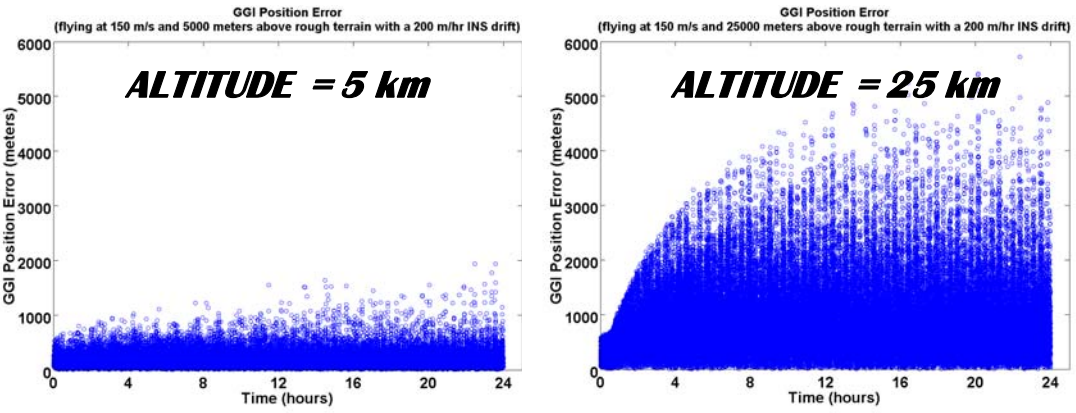


Figure 42: Altitude's Effect on GGI Solutions on an ISR Mission

Optimistic and Pessimistic Performance Perspectives

The optimistic and pessimistic scenarios intend to offer a best-case/worst-case perspective on GAME performance for a general aircraft. Both scenarios employ GGIs and maps with 0.1 and 0.01 Eö of noise, respectively, implying these simulations offer a look at GAME performance a decade or more from today. The optimistic scenario flies at 5 kilometers over rough terrain and uses all five components of the gravity gradient tensor, while the pessimistic scenario flies at 15 kilometers over smooth terrain and uses only three components. Table 11summarizes the conditions.

Table 11: Variables for Optimistic and Pessimistic Simulations

Variable	The Optimist	The Pessimist
Terrain	Rough	Smooth
Altitude (km)	5	15
Velocity (m/s)	150	150
Flight Duration (hr)	2, 4, 8, 16	2, 4, 8, 16
INS Drift Rate (m/hr)	20, 200, 2000	20, 200, 2000
Position Update Rate	1	1
GGI Components	5	3
GGI Noise (Eö)	0.1	0.1
Map Noise (Eö)	0.01	0.01
Map Resolution (arcseconds)	3	3
Map Interpolation (arcseconds)	3	3

The figures on the following page present the performance gains for the optimistic and pessimistic scenarios, while allowing the INS drift rate and flight duration to vary. This approach provides the potential for returns on investments in a GGI with 0.1 Eö of noise, given the aircraft's INS drift rate and flight duration. From the optimist's

perspective, a GGI noise capability of 0.1 Eö generally offers excellent potential for returns on investment when coupled with an INS that drifts 2 kilometers per hour. Good returns are expected with an INS that drifts 200 meters per hour, given flight durations longer than about 4 hours. Unfortunately, the pessimist's perspective indicates that a good potential for returns only exists for flights longer than 4 hours with an INS that drifts 2 kilometer per hour. The potential for returns grows significantly with INS drift rates more than 200 meters per hour and for longer flight durations. Taken together, the optimistic and pessimistic scenarios suggest that a 0.1 Eö GGI has an excellent to good potential for returns on investments with an INS that drifts 2 kilometers per hour. Good to poor potential exists with an INS that drifts 200 meters per hour, although long endurance missions would still benefit from GAME even under the pessimist's worst-case conditions. Most of these performance gains would receive a mild boost with interpolation applied in the map matching algorithm.

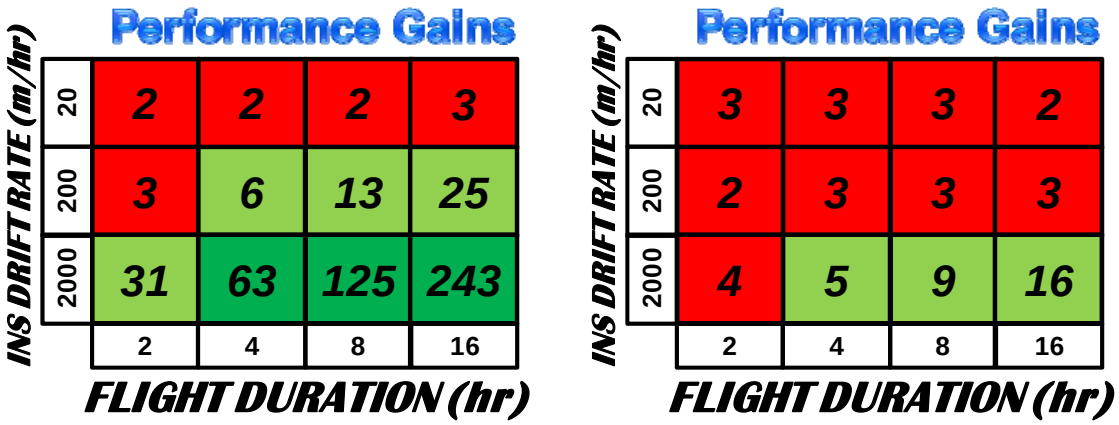


Figure 43: The Optimist (*left*) and Pessimist (*right*)

V. Conclusions

GAME and Aircraft Navigation

At default conditions, GAME performs 35 times better than an unaided INS, with a break even point of 4 minutes. Given information from the INS, gravity gradiometry and map matching achieves a CEP of 141 meters. Thanks to the Kalman filter and good estimates of uncertainties, GAME outperforms both the GGI and unaided INS, achieving a CEP of 128 meters. Granted, the default conditions are optimistic in some respects, but all the simulations, covering wide ranges of conditions, generally show that GAME positively affects navigation performance. Quality methods for bringing together navigation information from multiple sources and calculating uncertainties ensure that GAME improves navigation performance, even when GGI position solutions are less accurate than the INS. The amount of improvement depends on many variables, and this paper only investigated 10 of them. Other variables, this paper's assumptions, and the limitations of this research effort leave caveats to be explored, some of which the final section discusses, including an understanding of GAME's maximum performance limits.

The results of the sensitivity analysis provide a fundamental understanding of how important variables affect GAME's performance. Combined effects of variables, other than those presented in this paper, can be estimated with information presented in this paper or additional simulations. Considering the terrains selected for this research, rough terrain provides 2 to 5 times more accurate position solutions than smooth terrain. GAME performance improves with lower altitudes and more frequent position updates. The algorithm's unsuccessful map matches at low altitudes suggest that this research does not provide enough information for conclusions about GAME's performance at altitudes

near the local terrain height. While the average terrain height might represent the optimal altitude for GAME performance, navigating in close proximity to large terrain features requires maps based on more accurate terrain data and robust algorithms. Velocity did not significantly affect performance, but the simulations did not model velocity's effects on GGI sensors. Based on Roger's research, a more realistic GGI model would show that velocity affects accuracy. Decreases in GGI and map noise improve performance, but must work in concert with map resolution, interpolation, and the capabilities of the map matching algorithm to attain full potential. An insignificant amount of information is lost when decreasing map resolution from 3 arcseconds to 48, but this cannot demonstrate how much information would be gained with map resolutions higher than 3 arcseconds. GAME performance improves as the number of components of the gravity gradient tensor increases up to five. However, the best value appears to use three components, assuming each additional component comes at a proportional increase in costs. Although performance gains increase with flight duration, the actual GGI solutions do not significantly change. This simply communicates that missions with longer flight durations have more time to enjoy the improved GAME solutions, relative to an unaided INS that drifts boundlessly. Similarly, performance gains increase when GAME couples with an INS with higher drift rates. However, the accuracy of the GGI solutions decreases with higher INS drift rates. This phenomenon relates to the coupling of the INS and the map matching algorithm, where lower INS uncertainties allow the algorithm to search smaller map areas. If the INS communicates a higher uncertainty, the algorithm searches a larger map area and possibly finds other probable locations, thus reducing the GGI solution's certainty or even resulting in a less accurate position solution.

The Conditions that Make GAME Feasible for Aircraft Navigation

Given a quality navigation computer and map matching algorithm, GAME generally delivers positive effects on navigation solutions, making it a feasible aircraft navigation aid in most scenarios. Benefits include the provision of discrete position information that typically hovers around a steady state accuracy, which is primarily dependent on the 10 variables discussed in this paper (i.e. terrain, altitude, velocity, flight duration, INS drift, position update rate, GGI components, GGI and map noise, map resolution, and interpolation). Furthermore, GAME offers its position information worldwide while preserving the unique strengths of an INS; namely, its passive, all-weather, and undeniable capabilities. Under some conditions, such as short flights, orbiting over flat terrain, flying at high altitudes, and working with a very accurate INS, GAME has a neutral effect on position accuracies. Taken to the extreme and rolled together with poor algorithms, GAME could harm navigation solutions, especially in the short term. In general, however, GAME positively affects navigation performance under most conditions, given a quality navigation computer and map matching algorithm.

From the perspective of worthwhile investments, the practical simulations and performance gains, supported by knowledge from the sensitivity analysis, point to the conditions that make GAME feasible for aircraft navigation. Results under conditions other than those presented in this paper can be estimated with information presented in this paper or additional simulations. With 1 Eö of GGI noise and 2,000 meters per hour of INS drift, a good to poor potential for returns should be expected for the cargo and ISR missions. As defined in this paper, these missions apply to many other scenarios, including long distance and long endurance. Examples include loitering, ISR, and long

range attack and transportation. Even better performance gains occur for these scenarios at low altitudes. This represents a level of performance within reach of today's technologies. Investment in GAME with only fighter missions in mind would provide a solidly poor potential for returns in the short term, although the capability might be a nice addition in today's fighter-like scenarios at no cost.

Looking at the near future and considering a GGI capable of 0.1 Eö noise levels, aircraft with INS drift rates greater than 200 meters per hour possess a solid potential for good returns under most conditions. For long distance or long endurance aircraft, GAME provides good potential even with a 200 meter per hour INS. Optimistic and pessimistic outlooks support these conclusions and provide a useful tool for estimating performance gains, given an investment that produces 0.1 Eö GGIs for aircraft navigation.

The sensitivity analysis shows that performance measures in this paper improve with a coordinated effort to reduce noise levels, increase map resolution, and improve interpolation and map matching algorithms. These four factors are intertwined, and the weakest link limits GAME performance despite improvements in the other areas. When these four factors improve simultaneously, GGI solution accuracy significantly improves across all conditions and, in turn, produces more accurate GAME solutions, increases performance gains, and lowers break even points. GAME might never achieve GPS-level accuracy, but it provides position updates with respectable accuracy, especially compared to other navigation aids. Given the Chief of Staff of the Air Force's insistence that Joint forces reduce GPS dependence, a GAME INS could lead the market for a next-generation navigation package. The GAME INS provides what no other aircraft navigation package can offer…passive, all-weather, and undeniable navigation information.

Future Research and Technologies that Will Improve GAME Performance

The Art of Map Making - Validating gravity gradient models stands as the most immediate action needed to support GAME. Gravity gradient maps represent the foundation for using gravity gradiometry and map matching as a navigation aid. If modeled maps do not accurately portray reality, then further research can only build on an uncertain foundation. Whether measured or modeled information builds the maps, conducting actual surveys characterizes the nature of true gravity gradients. Surveys might quantify model inaccuracies, show how much gravity gradients change with time, identify poorly modeled locations, and verify the important frequencies of gravity gradient information. These types of validations provide answers to many questions, including, what methods result in the best maps? Should the maps be modeled, surveyed, or some combination? What's the optimal map resolution? How does GAME perform in other parts of the world, especially considering mountains, desserts, oceans, and extreme latitudes? What does it take for GAME to work well at low altitudes; near or within cities? Surveys provide the validations needed to answer these questions, which ultimately ensure that future research and investments build on a strong foundation.

The question regarding whether maps should be modeled or surveyed bears further discussion. A refined survey that addresses the frequencies of information available in gravity gradients, as well as general sampling techniques around the world, helps determine whether models or surveys make the optimal maps. Optimal in this case refers to maps that meet the user's needs at the least cost or provide the best value for the investment. Furthermore, general and refined surveys help determine whether full surveys of the Earth are required, or maybe just in some locations. If surveys match the

models within acceptable accuracies, modeled maps might be good enough, or maybe limited surveys can adequately improve models. For example, surveys might identify biases and lead to convenient correction factors. Further efforts might also identify the most efficient methods for including terrain effects in map models (e.g. how much terrain to include, what data resolution to use, and what data can be neglected at high altitude). Surveys over time also determine how often new surveys should be performed and whether updates to maps due to large mass movements can be made with calculations or require new surveys. Once again, without such validations, predictions of GAME's performance rest on a foundation that's only as solid as the models used in the predictions.

Strengthening the GGI – The next critical step in achieving GAME's potential is to support improvements in GGI technology. Although the demand for lower noise levels will probably never be satiated, most of this paper's simulations focused on GGIs with noise levels of 1, 0.1, and 0.01 Eö. While today's technologies are within reach of 1 Eö, further advancements would be necessary to ensure that noise levels could be held down onboard an aircraft, in a smaller and lighter package, and achieving all the necessary integrations with the aircraft. Needless to say, 0.1 Eö noise levels are even further into the future and will require all of the previously mentioned efforts and more. Additionally, GGIs should efficiently measure at least three components of the gravity gradient tensor, including the Γ_{DD} component, which could also assist the INS with estimating the gravity vector. Navigation computers require all of these features at high data rates, with accurate filtering and time stamping of data.

Smarter Algorithms – Finally, investment efforts must develop the brains of the GAME. While many algorithms already provide quality map matching services, GAME demands special attention in some areas. First, the nature of gravity gradients make many map matching methods applicable to GAME. Selecting the optimal method requires careful consideration, including the possibility that different algorithms provide the best performance at different times. For example, the uniqueness of gravity gradients throughout the world offers the potential for GAME to provide position solutions with no prior information. This process differs significantly from a map matching algorithm that receives assistance from and tracks along with an INS. While the likelihood function works well for tracking, a particle mass filter might work better for initializing a position, with little to no prior information, and take advantage of map features and worldwide patterns such as found with Γ_{ND} in Figure 11 on page 34. A third algorithm might even be better suited for taking advantage of high speed flight by sensing and identifying large features or landmarks on the maps, and avoiding intense algorithms that attempt to process every byte of data. Finally, certain algorithms might offer capabilities to adapt to static and transient biases in gravity gradients, such as those caused by changes in aircraft configuration, flying near other aircraft, variations in atmospheric conditions, large mass movements on earth, and other changes in gravity gradients. Thus, multiple algorithms for GAME is a powerful option.

From the perspective of the simple map matching algorithm used in this paper, many things could be done to improve GAME. First, and most importantly, altitude must be added to the map matching algorithm as an unknown. The addition of another unknown could demand changes to the algorithm's logic and significantly affect

GAME's performance measures. The efficiency and accuracy of the algorithm could be improved so that it initially searches a small area for position solutions (e.g. 1-standard deviation of the estimated INS error), and then incrementally widens the search if a position solution is not found with an acceptable uncertainty.

Simulations in this research effort also encountered situations where the likelihood function assigns insignificant likelihoods to all position solution candidates, despite low noise levels. This essentially results in a rejection of all position solution candidates, because the differences between measured and expected (i.e. senor and map) values are too large, or similar at multiple locations. Computer processing capacity and limitations on interpolation also contributed to the inability of the map matching algorithm to identify unique position solutions. In some scenarios, interpolation decreased the accuracy of position solutions. These phenomena prevent this analysis from driving the critical variables, such as noise levels, interpolation, and map resolutions, to the maximum GAME performance limits. Thus, the simulations in this analysis did not explore the full potential of GAME position accuracies, although the physical attainment of such accuracies is probably far into the future. The simulations, however, show that GAME improves navigation accuracy under almost all conditions. However, achieving a feasible return on investment with today's technologies may be limited to long distance and long endurance missions. Depending on the relative advancement of INS and GGI sensors, tomorrow's technologies promise even greater returns on GAME investments, paving the way for GAME to potentially dominate the market for secure and covert navigation aids that preserve the strengths of aircraft inertial navigation systems.

Appendix A. MATLAB Computer Program

The following computer program, written in MATLAB 2008b, ran all of the simulations presented in this paper. The program simulates an aircraft inertial navigation system enhanced with gravity gradiometry and map matching (GAME). Section III of this paper, *Methodology*, describes the program in more detail. Appendix B presents the full table of results. Due to their size, the figures presented in previous sections are the only plots presented in this paper. Before running this program, gravity gradient maps must be stored in accordance with the instructions in the program. For this paper, a modified version of the program developed by Rogers[19] created the maps.

Minor modifications can be made to this program to accommodate different maps and resolutions, aircraft starting points and flightpaths, and INS drifts. Although the program was written with other options in mind, the addition of aircraft dynamics, the inclusion of altitude as an unknown, the detailed modeling of INS and GGI sensors, and the choice of different map matching algorithms require more significant modifications. In these circumstances, computer programmers could work with this program, but might consider developing a new program and using this program as a guide.

```
%*************************************************************************%
%  Anthony DeGregoria, Air Force Institute of Technology, March 2010
%
%  This program simulates an aircraft inertial navigation system enhanced
%     with gravity gradiometry and map matching (GAME).
%
%  NOTES:
%  - Before running this program, gravity gradient maps must be stored
%       in accordance with the instructions starting on line 58 and updates
%       made to the section starting on line 31
%  - This program treats the stored gravity gradient maps as truth data
%  - User inputted noise is added to the truth maps
%  - The aircraft flies a hard coded flightpath, which can be modified in
%       the section starting on line 125
%  - The aircraft flies at a constant velocity and altitude
%  - The INS always drifts southeast.
%       Changes can be made in the section starting on line 185
%  - The Likelihood Function sums the inputted GGI & map noise
%*************************************************************************%

close all
clear all
clc
```

```
%Define some constants
    G=6.67E-11; %Universal Gravity Constant
    a=6378137; %semi-major axis of the Earth in meters
    e2=0.0818191908426^2; %Earth's eccentricity squared
    Eotvos=1E-9; %used to convert units to Eotvos

%The aircraft's true starting position and the resolution & reference vector for the truth maps are hard coded here

    M=input('Rough or Smooth Terrain [R/S]? ','s');
    if ((M=='R')|(M=='r')) %this section provides information to the algorithm about the rough maps
        position_start=[35.86 -121.32];  %[lat long] of starting point for rough terrain true trajectory;
                                         %Geodetic coordinates (WGS84 reference ellipsoid)
        Res_source=1; %put the Resolution Level of the truth maps here
        GGIsourcemap_resolution=3*2^(1-Res_source); %resolution of source map (arcseconds)
        refvec=[1200, 37.0004, -122.0004]; %information pertaining to (row 1, column 1) of rough terrain gradient maps
                                           %refvec=[relates to grid size, lattitude, longitude]
        if (M=='r')
            M='R';
        end
    elseif ((M=='S')|(M=='s')) %this section provides information to the algorithm about the smooth maps
        position_start=[35.86 -89.32];  %[lat long] of starting point for smooth terrain true trajectory;
                                        %Geodetic coordinates (WGS84 reference ellipsoid)
        Res_source=1; %put the Resolution Level of the source/truth maps here
        GGIsourcemap_resolution=3*2^(1-Res_source); %resolution of source map (arcseconds)
        refvec=[1200, 37.0004, -90.0004]; %information pertaining to (row 1, column 1) of smooth terrain gradient maps;
                                          %refvec=[relates to grid size, lattitude, longitude]
        if (M=='s')
            M='S';
        end
    else
        fprintf('Wrong Answer!'); break;
    end

%User inputs altitude and computer loads the corresponding gravity gradient maps for the selected terrain and alitude;
    %the maps must be stored in a sub-directory named "GradientMaps"; files must follow the naming convention xTxxFL##,
    %where the first x designates the terrain ('R' for rough or 'S' for smooth), the second and third x designate the
    %component of the gravity gradient tensor (e.g. 'xz'), and the two # signs designate the flight level in kilometers
    %(e.g. '05' designates 5,000 meters altitude and '30' designates 30,000 meters altitude)
    altitude=input('Enter Altitude (height above the average terrain height in meters) = ');
    altitude_km=round(altitude/1000);
    if altitude_km<10
        altitude_km=['0' int2str(altitude_km)];
    else
        altitude_km=int2str(altitude_km);
    end
    map_file_Txx=[M 'TxxFL' altitude_km]; %loads the Txx components of the gravity gradient map
    map_Txx=load (['GradientMaps\' map_file_Txx]);
    map_file_Txy=[M 'TxyFL' altitude_km]; %loads the Txy components of the gravity gradient map
    map_Txy=load (['GradientMaps\' map_file_Txy]);
    map_file_Txz=[M 'TxzFL' altitude_km]; %loads the Txz components of the gravity gradient map
    map_Txz=load (['GradientMaps\' map_file_Txz]);
    map_file_Tyz=[M 'TyzFL' altitude_km]; %loads the Tyz components of the gravity gradient map
    map_Tyz=load (['GradientMaps\' map_file_Tyz]);
    map_file_Tzz=[M 'TzzFL' altitude_km]; %loads the Tzz components of the gravity gradient map
    map_Tzz=load (['GradientMaps\' map_file_Tzz]);

%User inputs velocity, INS/GGI performance factors, map information, and filename for outputs
    velocity=input('Enter Aircraft Velocity (meters per second) = ');
    INS_drift=input('Enter INS Drift Rate (CEP50 in meters per hour) = ');
    update_rate_GGI=input('Enter GGI Data Rate (seconds) = ');
    update_rate_GAME=input('Enter how often you would like the navigation computer to run GAME (seconds) = ');
    GGIsignal_noise=input('Enter GGI Noise Level\n   (1 standard deviation measured in Eotvos) = ');
    GGImap_noise=input('Enter Gravity Gradient Map Noise Level\n   (1 standard deviation measured in Eotvos) = ');
    Res_sim=input('Enter the simulated Gravity Gradient Map Resolution\n   (Resolution Level must be equal or less than
the database maps) enter...\n   "1" for 3 arcseconds,\n   "0" for 6 arcseconds,\n   "-1" for 12 arcseconds...? ');
        spacing=2^( Res_source- Res_sim);
    Res_interp=input('Enter how much you would like the algorithm to Interpolate the simulated maps\n   (Resolution
Level must be equal or greater than the database map) enter...\n   "1" for 3 arcseconds,\n   "2" for 1.5 arcseconds,\n
"3" for 0.75 arcseconds...? ');
        num_interps= Res_interp- Res_sim;
        GGImap_resolution=3*2^(1- Res_interp); %Gravity Gradient Maps will be interpolated from the
                                               %Res_sim resolution to this resolution (arcseconds)
    time_sim=input('Enter how long you would like the simulation to run (hours) = '); %75000*8*2/velocity/3600;
    filename=input('Enter a Filename for the results to be published = ','s');

%Set initial conditions for variables
    time=0; %the simulation starts at this time
    time_step=1; %defines the time step for the simulation in seconds
    %initial true position [lat, long, altitude, time]
        position_true(1,:)=[position_start(1,1), position_start(1,2), altitude, time];
    %row location of initial true position on map matrix
        position_true_row=floor((refvec(1,1)*2+1)-(refvec(1,2)-position_true(1,1))*3600/GGIsourcemap_resolution);
    %column location of initial true position on map matrix
        position_true_column=floor(-(refvec(1,3)-position_true(1,2))*3600/GGIsourcemap_resolution)+1;
    GGIsignal_true(1,1)=map_Txx.(map_file_Txx)(position_true_row,position_true_column); %Txx at initial location
    GGIsignal_true(1,2)=map_Txy.(map_file_Txy)(position_true_row,position_true_column); %Txy at initial location
    GGIsignal_true(1,3)=map_Txz.(map_file_Txz)(position_true_row,position_true_column); %Txx at initial location
    GGIsignal_true(1,4)=map_Tyz.(map_file_Tyz)(position_true_row,position_true_column); %Tyz at initial location
    GGIsignal_true(1,5)=map_Tzz.(map_file_Tzz)(position_true_row,position_true_column); %Tzz at initial location
    GGIsignal_INS(1,1:5)=GGIsignal_true(1,1:5); %gravity gradients at initial INS position
    P=zeros(3,3); %matrix whose diagonals represent the uncertainty of the INS position
    H=eye(3,3);
    position_Kalman(1,:)=position_true(1,:); %initial integrated navigation solution set to match true position
    position_INS(1,:)=position_true(1,:); %initial INS solution set to match true position
```

```
    position_GGI(1,:)=position_true(1,:); %initial GGI solution set to match true position
    num_matches_failed=0; %initial number of bad map matches set to zero
    num_matches_successful=0; %intial number of times that a map match is attempted
    BAILOUT=0; %a flag that breaks the map matching algorithm (GAME) loop
    track_angle=0; %initial direction the aircraft flies, measured in degrees with zero degrees pointing East

tic %start a clock to record how long it takes to accomplish the simulation
time_stamp=clock; %record the date & time of this simulation
h = waitbar(0,'The GAME is this percent complete...'); %tells user what percent of the simulation is complete

while BAILOUT<1 %stops the simulation once the bailout flag is set to one or graeter

    time=time+time_step; %propagates time
    fraction_complete=time/floor(time_sim)/3600; %calculates how much of the simulation is complete
    waitbar(fraction_complete,h) %updates the waitbar

%Model True Trajectory (position_true=[lattitude, longitude, altitude, time])
    %if you want to fly a circle pattern over the terrain, use this if statement


    %if you want to fly a square pattern over the terrain, use this if statement
    %if rem(time,floor(750000/velocity))==0
    %    track_angle=track_angle+90;
    %end

    %if you want to fly a star pattern over the terrain, use this if statement
     if rem(time,floor(75000/velocity))==0 %each segment of the start is 75 km long
         track_angle=track_angle+135; %repetitive 135-degree left turns make a start pattern
     end

    %Calculate Earth's radius parameters in meters (Dr. Raquet's EENG 533 class notes, AFIT)
    Rm=a*(1-e2)/(1-e2*sind(position_true(time,1))*sind(position_true(time,1)))^(3/2);
    Rp=a/(1-e2*sind(position_true(time,1))*sind(position_true(time,1)))^(1/2);

    %Calculate the change in latt, long, and altitude based on user input
    delta_latt_true=180/pi()*velocity*sind(track_angle)*time_step/(Rm+position_true(time,3));

delta_long_true=180/pi()*velocity*cosd(track_angle)*time_step/(Rp+position_true(time,3))/cosd(position_true(time,1));
    delta_alti_true=0*time_step; %zero may be replaced by an ascent/descent rate specificed by the user

    %Propagate the aircraft's true position
    position_true(time+1,1)=position_true(time,1)+delta_latt_true;
    position_true(time+1,2)=position_true(time,2)+delta_long_true;
    position_true(time+1,3)=position_true(time,3)+delta_alti_true;
    position_true(time+1,4)=time;

%Record GGI Signal at True Location, using linear interpolation and including the random noise specified by the user

    position_true_row=floor( (refvec(1,1)*2+1) - (refvec(1,2)-position_true(time+1,1))*3600/GGIsourcemap_resolution  );
    position_true_column=floor( -(refvec(1,3)-position_true(time+1,2)) *3600/GGIsourcemap_resolution)+1;

    GGIsignal_true_mapTxx=map_Txx.(map_file_Txx)...
        (position_true_row:position_true_row+1, position_true_column:position_true_column+1);
    GGIsignal_true_mapTxy=map_Txy.(map_file_Txy)...
        (position_true_row:position_true_row+1, position_true_column:position_true_column+1);
    GGIsignal_true_mapTxz=map_Txz.(map_file_Txz)...
        (position_true_row:position_true_row+1, position_true_column:position_true_column+1);
    GGIsignal_true_mapTyz=map_Tyz.(map_file_Tyz)...
        (position_true_row:position_true_row+1, position_true_column:position_true_column+1);
    GGIsignal_true_mapTzz=map_Tzz.(map_file_Tzz)...
        (position_true_row:position_true_row+1, position_true_column:position_true_column+1);

    lat=refvec(1,2)+GGIsourcemap_resolution/3600*(position_true_row-refvec(1,1)*2-1);
    lon=refvec(1,3)+(position_true_column-1)*GGIsourcemap_resolution/3600;

    latt=[lat+GGIsourcemap_resolution/3600, lat];
    long=[lon, lon+GGIsourcemap_resolution/3600];

    GGIsignal_true(time+1,1)=normrnd(interp2(latt, long, GGIsignal_true_mapTxx,...
        position_true(time+1,1), position_true(time+1,2)), GGIsignal_noise);
    GGIsignal_true(time+1,2)=normrnd(interp2(latt, long, GGIsignal_true_mapTxy,...
        position_true(time+1,1), position_true(time+1,2)), GGIsignal_noise);
    GGIsignal_true(time+1,3)=normrnd(interp2(latt, long, GGIsignal_true_mapTxz,...
        position_true(time+1,1), position_true(time+1,2)), GGIsignal_noise);
    GGIsignal_true(time+1,4)=normrnd(interp2(latt, long, GGIsignal_true_mapTyz,...
        position_true(time+1,1), position_true(time+1,2)), GGIsignal_noise);
    GGIsignal_true(time+1,5)=normrnd(interp2(latt, long, GGIsignal_true_mapTzz,...
        position_true(time+1,1), position_true(time+1,2)), GGIsignal_noise);

%Model INS trajectory and uncertainty

    %The user inputted INS drift is split into two equal components in the lattitudinal and longitudinal directions
    delta_latt_INS=180/pi()*(INS_drift/sqrt(2)/3600*time_step)/(Rm+mean(position_true(time:time+1,3)));
    delta_long_INS=180/pi()*(INS_drift/sqrt(2)/3600*time_step)/(Rp+mean(position_true(time:time+1,3)))/...
        cosd(mean(position_true(time:time+1,1)));
    delta_alti_INS=0;

    %The new INS position is equal to the old position, plus the sensed aircraft
    %movement since the last position, and the INS drift, which is applied
    %in the south and east directions
    position_INS(time+1,1)=position_INS(time,1)+delta_latt_true-delta_latt_INS;
    position_INS(time+1,2)=position_INS(time,2)+delta_long_true+delta_long_INS;
    position_INS(time+1,3)=position_INS(time,3)+delta_alti_true+delta_alti_INS;
    position_INS(time+1,4)=time;
```

```
    %Record the gravity gradients along the INS flightpath
    position_INS_row=floor((refvec(1,1)*2+1)-(refvec(1,2)-position_INS(time+1,1))*3600/GGIsourcemap_resolution);
    position_INS_column=floor(-(refvec(1,3)-position_INS(time+1,2))*3600/GGIsourcemap_resolution)+1;

    GGIsignal_INS_mapTxx=map_Txx.(map_file_Txx)...
        (position_true_row:position_true_row+1,position_true_column:position_true_column+1);
    GGIsignal_INS_mapTxy=map_Txy.(map_file_Txy)...
        (position_true_row:position_true_row+1,position_true_column:position_true_column+1);
    GGIsignal_INS_mapTxz=map_Txz.(map_file_Txz)...
        (position_true_row:position_true_row+1,position_true_column:position_true_column+1);
    GGIsignal_INS_mapTyz=map_Tyz.(map_file_Tyz)...
        (position_true_row:position_true_row+1,position_true_column:position_true_column+1);
    GGIsignal_INS_mapTzz=map_Tzz.(map_file_Tzz)...
        (position_true_row:position_true_row+1,position_true_column:position_true_column+1);

    lat=refvec(1,2)+GGIsourcemap_resolution/3600*(position_true_row-refvec(1,1)*2-1);
    lon=refvec(1,3)+(position_true_column-1)*GGIsourcemap_resolution/3600;

    latt=[lat+GGIsourcemap_resolution/3600, lat];
    long=[lon, lon+GGIsourcemap_resolution/3600];

    GGIsignal_INS(time+1,1)=interp2(latt, long, GGIsignal_true_mapTxx,...
        position_true(time+1,1), position_true(time+1,2));
    GGIsignal_INS(time+1,2)=interp2(latt, long, GGIsignal_true_mapTxy,...
        position_true(time+1,1), position_true(time+1,2));
    GGIsignal_INS(time+1,3)=interp2(latt, long, GGIsignal_true_mapTxz,...
        position_true(time+1,1), position_true(time+1,2));
    GGIsignal_INS(time+1,4)=interp2(latt, long, GGIsignal_true_mapTyz,...
        position_true(time+1,1), position_true(time+1,2));
    GGIsignal_INS(time+1,5)=interp2(latt, long, GGIsignal_true_mapTzz,...
        position_true(time+1,1), position_true(time+1,2));

    %Calculate and propagate the uncertainty of the INS
    variance=(INS_drift/3600*time_step*1.2)^2; %convert INS drift into a variance (meters)^2
    variance_latt_INS=(180/pi()*sqrt(variance)/(Rm+mean(position_true(time:time+1,3))))^2; %(degrees^2)
    variance_long_INS=(180/pi()*sqrt(variance)/(Rp+mean(position_true(time:time+1,3)))/...
        cosd(mean(position_true(time:time+1,1))))^2; %(degrees^2)
    variance_alti_INS=.1; %arbitrary value...currently does not affect navigation solutions (meters^2)
    P=P+[variance_latt_INS,0,0;0,variance_long_INS,0;0,0,variance_alti_INS]; %propagate uncertainties forward in time

%Find best location on GGI map that matches GGI signal and calculate its uncertainty

    if rem(time,update_rate_GAME)==0 %only attempts GAME as frequently as the user specified

    %Capture the portion of the map representing the 3-sigma uncertainty of the INS position

        sigma3=[3*sqrt(P(1,1)*time), 3*sqrt(P(2,2)*time), 3*sqrt(P(3,3)*time)].*3600./GGIsourcemap_resolution;
        sigma3=round(sigma3);
        if sigma3(1,1)<spacing %signal_candidates_sim must be minimum of 2-by-2 grid or code crashes on interp2 command
            sigma3(1,1)=spacing;
        end
        if sigma3(1,2)<spacing
            sigma3(1,2)=spacing;
        end
        if sigma3(1,1)/spacing*2^num_interps<4 %signal_candidates must be minimum 9-by-9 grid or code crashes on interp
            sigma3(1,1)=4*spacing/2^num_interps;
        end
        if sigma3(1,2)/spacing*2^num_interps<4
            sigma3(1,2)=4*spacing/2^num_interps;
        end
        n=(sigma3(1,1)*2+1)*(sigma3(1,2)*2+1);

        %Grab a portion of the truth map, but at the user's requested resolution (i.e. Res_sim)
        clear signal_candidates; clear signal_candidates_sim;
        signal_candidates_sim(:,:,1)=map_Txx.(map_file_Txx)...
            (position_INS_row-sigma3(1,1):spacing:position_INS_row+sigma3(1,1),...
            position_INS_column-sigma3(1,2):spacing:position_INS_column+sigma3(1,2));
        signal_candidates_sim(:,:,2)=map_Txy.(map_file_Txy)...
            (position_INS_row-sigma3(1,1):spacing:position_INS_row+sigma3(1,1),...
            position_INS_column-sigma3(1,2):spacing:position_INS_column+sigma3(1,2));
        signal_candidates_sim(:,:,3)=map_Txz.(map_file_Txz)...
            (position_INS_row-sigma3(1,1):spacing:position_INS_row+sigma3(1,1),...
            position_INS_column-sigma3(1,2):spacing:position_INS_column+sigma3(1,2));
        signal_candidates_sim(:,:,4)=map_Tyz.(map_file_Tyz)...
            (position_INS_row-sigma3(1,1):spacing:position_INS_row+sigma3(1,1),...
            position_INS_column-sigma3(1,2):spacing:position_INS_column+sigma3(1,2));
        signal_candidates_sim(:,:,5)=map_Tzz.(map_file_Tzz)...
            (position_INS_row-sigma3(1,1):spacing:position_INS_row+sigma3(1,1),...
            position_INS_column-sigma3(1,2):spacing:position_INS_column+sigma3(1,2));

        for ii=1:1:5
            %Noise is added to the gravity gradient maps to simulate inaccuracies in the stored data;
            signal_candidates_sim(:,:,ii)=normrnd(signal_candidates_sim(:,:,ii), GGImap_noise);

            %The gravity gradient maps are interpolated to the user's requested resolution (i.e. Res_interp)
            signal_candidates(:,:,ii)=interp2(signal_candidates_sim(:,:,ii), num_interps, 'linear');

    %Calculate the point on the map with the maximum likelihood of matching the GGI sensor data
            signal_candidates(:,:,ii)=(signal_candidates(:,:,ii)-GGIsignal_true(time+1,ii)).^2;
        end

        signal_candidates(:,:,6)=sum(signal_candidates(:,:,1:5),3);
        signal_candidates(:,:,6)=exp(-signal_candidates(:,:,6)./2./((GGIsignal_noise+GGImap_noise)^2))./...
```

```
            sqrt((2*pi())^5*(GGIsignal_noise+GGImap_noise)^2);

        likelihood=max(max(signal_candidates(:,:,6)));
        [signal_candidates_row, signal_candidates_column]=find(signal_candidates(:,:,6)==likelihood);

        position_GGI_row=floor((signal_candidates_row-1)/2^( Res_interp-1))+position_INS_row-sigma3(1,1)-1;
        position_GGI_column=floor((signal_candidates_column-1)/2^( Res_interp-1))+position_INS_column-sigma3(1,2)-1;

    if length(signal_candidates_row)==1 %when there is not exactly one location with the maximum likelihood(e.g. zero),
            %the algorithm will set the map matching solution equal to the current INS position
        %The formula for converting matrix indices to positions needs to be updated,
            %if flying over locations with negative lattitude and/or positive longitude
        position_GGI(time+1,1)=refvec(1,2)+3/3600*(position_GGI_row-refvec(1,1)*2-1)+...
            rem(signal_candidates_row-1,2^( Res_interp-1))*GGImap_resolution/3600;
        position_GGI(time+1,2)=refvec(1,3)+(position_GGI_column-1)*3/3600+...
            rem(signal_candidates_column-1,2^( Res_interp-1))*GGImap_resolution/3600;
        position_GGI(time+1,3)=position_INS(time+1,3);
        position_GGI(time+1,4)=time;

        %Calculate magnitude of GGI error in meters
            %position_error_GGI=[north error, east error, up error, time, total error magnitude] (measured in meters)
            num_matches_successful=num_matches_successful+1;
            position_error_GGI(num_matches_successful,1:3)=position_true(time+1,1:3)-position_GGI(time+1,1:3);
            %convert INS lattitude and longitude error from degrees to meters
            position_error_GGI(num_matches_successful,1)=pi()/180*position_error_GGI(num_matches_successful,1)*Rm;
            position_error_GGI(num_matches_successful,2)=pi()/180*position_error_GGI(num_matches_successful,2)*...
                (Rp+position_true(time+1,3))*cosd(position_true(time+1,2));
            position_error_GGI(num_matches_successful,4)=time;
            position_error_GGI(num_matches_successful,5)=sqrt(position_error_GGI(num_matches_successful,1).^2+...
                position_error_GGI(num_matches_successful,2).^2); %magnitude of INS RMS position error in meters

        %Calculate uncertainty associated with point on map with maximum likelihood (modified from Capt William Storms)
        pdf_x=signal_candidates(signal_candidates_row,:,6);
        pdf_y=signal_candidates(:,signal_candidates_column,6);
        r=round(1000/size(signal_candidates,1));
        s=round(1000/size(signal_candidates,2));
        pdf_x=interp(pdf_x,s);
        pdf_y=interp(pdf_y,r);
        pdf_x=pdf_x./sum(pdf_x);
        pdf_y=pdf_y./sum(pdf_y);
        ii=1; jj=size(pdf_x,2); kk=1; ll=size(pdf_y,1);
        pdf_x_sum_left=0; pdf_x_sum_right=0;pdf_y_sum_left=0; pdf_y_sum_right=0;
        while pdf_x_sum_left<0.16
            pdf_x_sum_left=pdf_x_sum_left+pdf_x(ii);
            ii=ii+1;
        end
        while pdf_x_sum_right<0.16
            pdf_x_sum_right=pdf_x_sum_right+pdf_x(jj);
            jj=jj-1;
        end
        while pdf_y_sum_left<0.16
            pdf_y_sum_left=pdf_y_sum_left+pdf_y(kk);
            kk=kk+1;
        end
        while pdf_y_sum_right<0.16
            pdf_y_sum_right=pdf_y_sum_right+pdf_y(ll);
            ll=ll-1;
        end
        sigma_x=ceil((jj-ii)/2)/s*GGImap_resolution/3600;
        sigma_y=ceil((ll-kk)/2)/r*GGImap_resolution/3600;
        %3sigma_x and 3sigma_y must be equal or greater than the resolution of the gravity gradient maps
        sigma_min=GGImap_resolution/3600/3;
        if sigma_x<sigma_min
            sigma_x=sigma_min;
        end
        if sigma_y<sigma_min
            sigma_y=sigma_min;
        end
        R=[sigma_x^2,0,0;0,sigma_y^2,0;0,0,20];

    else
            clear signal_candidates_row; clear signal_candidates_column; clear position_GGI_row; clear
position_GGI_column;
            position_GGI(time+1,:)=position_INS(time+1,:); %GGI accepts INS positin solution when map matching fails
            R=[1.78e14,0,0;0,1.78e14,0;0,0,20]; %high uncertainties minimize weight of bad solution in Kalman filter
            num_matches_failed=num_matches_failed+1; %counts the number of failed map matches
    end

    %Update Kalman and INS position...using a discrete linear Kalman Filter!
        K=P*H'/(H*P*H'+R);
        position_Kalman(time+1,1:3)= (position_INS(time+1,1:3)' + K*(position_GGI(time+1,1:3)'-...
            H*position_INS(time+1,1:3)'))';
        position_Kalman(time+1,4)=time;
        position_INS(time+1,:)=position_Kalman(time+1,:);
        P=(eye(3,3)-K*H)*P;

    else

        position_Kalman(time+1,:)=position_INS(time+1,:); %when no map match is attempted, Kalman equals INS solution

    end

%Calculate magnitude of INS error in meters
    %position_error_INS=[north error, east error, up error, time, total error magnitude] (measured in meters)
```

```
    position_error_INS(time+1,1:3)=position_true(time+1,1:3)-position_INS(time+1,1:3);
    %convert INS lattitude and longitude error from degrees to meters
    position_error_INS(time+1,1)=pi()/180*position_error_INS(time+1,1)*Rm;
    position_error_INS(time+1,2)=pi()/180*position_error_INS(time+1,2)*...
        (Rp+position_true(time+1,3))*cosd(position_true(time+1,2));
    position_error_INS(time+1,4)=time;
    %magnitude of INS RMS position error in meters
    position_error_INS(time+1,5)=sqrt(position_error_INS(time+1,1).^2+position_error_INS(time+1,2).^2);

%If you get too close to the edge of the map, then stop the simulation
    if time==round(time_sim*3600)
        BAILOUT=100;
    end
    if (position_true(time+1,1)>(refvec(1,2)-.4) | position_true(time+1,1)<(refvec(1,2)-1.6) |...
            position_true(time+1,2)<(refvec(1,3)+0.4) | position_true(time+1,2)>(refvec(1,3)+1.7))
        BAILOUT=100;
        fprintf('You flew too close to the edge of the map!');
    end
    if (position_INS(time+1,1)>(refvec(1,2)-.1) | position_INS(time+1,1)<(refvec(1,2)-1.9) |...
            position_INS(time+1,2)<(refvec(1,3)+0.1) | position_INS(time+1,2)>(refvec(1,3)+1.9))
        BAILOUT=100;
        fprintf('Your INS drifted too close to the edge of the map!');
    end

end

processor_time=toc;
waitbar(1,h,'Calculating Performance and Saving Results...')

%Document the results of the simulation

    %Calculate accuracy of GAME position solutions
        position_error_GAME_RMSmean=mean(position_error_INS(:,5));
        position_error_GAME_RMSstd=std(position_error_INS(:,5));
        position_error_GAME_CEP50=position_error_GAME_RMSmean/1.2;
        position_error_GAME_CEP50true=median(position_error_INS(:,5));

        fprintf('%g meters:  GAME mean RMS horizontal position error during the flight.\n',position_error_GAME_RMSmean)
        fprintf('%g meters:       standard deviation.\n', position_error_GAME_RMSstd)
        fprintf('%g meters:       equivalent CEP50.\n', position_error_GAME_CEP50)
        fprintf('%g meters:       actual CEP50.\n\n', position_error_GAME_CEP50true)

    %Calculate accuracy of GGI position solutions
        position_error_GGI_RMSmean=mean(position_error_GGI(:,5));
        position_error_GGI_RMSstd=std(position_error_GGI(:,5));
        position_error_GGI_CEP50=position_error_GGI_RMSmean/1.2;
        position_error_GGI_CEP50true=median(position_error_GGI(:,5));

        fprintf('%g meters:  GGI mean RMS horizontal position error during the flight.\n', position_error_GGI_RMSmean)
        fprintf('%g meters:      standard deviation.\n', position_error_GGI_RMSstd)
        fprintf('%g meters:      equivalent CEP50.\n', position_error_GGI_CEP50)
        fprintf('%g meters:      actual CEP50.\n\n', position_error_GGI_CEP50true)

    %Calculate the Performance Gain
        if (INS_drift*time_sim)>position_error_GAME_CEP50
            performance_gain=INS_drift*time_sim/position_error_GAME_CEP50true;
        else
            performance_gain=-(position_error_GAME_CEP50true/INS_drift*time_sim);
        end
        fprintf('%g :  Your performance gain for playing the GAME was a factor of...\n', performance_gain)

    %Calculate the Break Even Point
        BEP=position_error_GAME_CEP50true/INS_drift*60; %break even point measured in minutes
        fprintf('%g :  The GAME will break even with the INS in this many minutes...\n\n', BEP)

    fprintf('%g of ', num_matches_failed)
    fprintf('%g attempts resulted in an unsuccessful map match.\n', num_matches_successful+num_matches_failed)

    %Write the results to a file
        FID=fopen(filename, 'a');
        fprintf(FID,'%s \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g
\t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \t %g \n',...
            M, altitude, velocity, time_sim, INS_drift, update_rate_GGI, update_rate_GAME, GGIsignal_noise,
GGImap_noise,...
            Res_source, Res_sim, Res_interp,...
            time_stamp(1), time_stamp(2), time_stamp(3), time_stamp(4), time_stamp(5), time_stamp(6),processor_time,...
            position_error_GAME_RMSmean, position_error_GAME_RMSstd, position_error_GAME_CEP50,
position_error_GAME_CEP50true,...
            position_error_GGI_RMSmean, position_error_GGI_RMSstd, position_error_GGI_CEP50,
position_error_GGI_CEP50true,...
            performance_gain, BEP, num_matches_failed, num_matches_successful+num_matches_failed);
        fclose('all');

%% PLOTS

    waitbar(1,h,'Generating Plots...')
    if M=='R'
        terrain='rough';
    elseif M=='S'
        terrain='smooth';
    else
        terrain=='unknown';
    end
    titl=['(flying at ' int2str(velocity) ' m/s and ' int2str(altitude) ' meters above ' terrain...
```

```
        ' terrain with a ' int2str(INS_drift) ' m/hr INS drift)'];

%Plot magnitude of INS error versus time
    figure
    plot(position_error_INS(:,4), position_error_INS(:,5))
    xlabel('Time (seconds)'); ylabel('GAME Position Error (meters)')
    title({'GAME Position Error versus Time'; titl}, 'FontSize', 16)

%Plot magnitude of GGI error versus time
    figure
    plot(position_error_GGI(:,4), position_error_GGI(:,5),'o')
    xlabel('Time (seconds)'); ylabel('GGI Position Error (meters)')
    title({'GGI Position Error versus Time'; titl}, 'FontSize', 16)

%Plot GGI signals along the true and INS flightpath versus time
    figure
    a=length(GGIsignal_INS)-1;
    plot((0:a)/60, GGIsignal_INS(:,5),'-k','LineWidth',2); hold on; plot((0:a)/60, GGIsignal_true(:,5),...
        ':k','LineWidth',2);
    xlabel('Time (minutes)', 'FontSize', 14); ylabel('Tzz (Eotvos)', 'FontSize', 14);
    legend('INS Flightpath','True Flightpath');
    title({'GGI Signals along True and INS Flightpaths'; titl}, 'FontSize', 16);

%Plot GGI signals along true and INS flightpaths versus time
    figure
    plot((0:a)/60, GGIsignal_INS(:,1),'-r'); hold on; plot((0:a)/60, GGIsignal_true(:,1),':r'); hold on;
    plot((0:a)/60, GGIsignal_INS(:,2),'-m'); hold on; plot((0:a)/60, GGIsignal_true(:,2),':m'); hold on;
    plot((0:a)/60, GGIsignal_INS(:,3),'-g'); hold on; plot((0:a)/60, GGIsignal_true(:,3),':g'); hold on;
    plot((0:a)/60, GGIsignal_INS(:,4),'-b'); hold on; plot((0:a)/60, GGIsignal_true(:,4),':b'); hold on;
    plot((0:a)/60, GGIsignal_INS(:,5),'-k'); hold on; plot((0:a)/60, GGIsignal_true(:,5),':k')
    xlabel('Time (minutes)', 'FontSize', 14); ylabel('Txx (Eotvos)', 'FontSize', 14);
    legend('INS Flightpath','True Flightpath');
    title({'GGI Signals along True and INS Flightpaths'; titl}, 'FontSize', 16);

%Plot the true and INS lattitudes and longitudes versus time
    figure
    [AX,H1,H2] = plotyy(position_true(:,4),position_true(:,1), position_true(:,4),position_true(:,2), 'plot');
    set(get(AX(1),'Ylabel'),'String','Lattitude')
    set(get(AX(2),'Ylabel'),'String','Longitude')
    set(H1,'Color','b')
    set(H2,'Color','g')
    axis(AX(1),[0 time 35 37])
    set(AX(1), 'YTick', [35 37]);
    if M=='R'
        axis(AX(2),[0 time -122 -120])
        set(AX(2),'YTick',[-122 -120]);
    else
        axis(AX(2),[0 time -90 -88])
        set(AX(2),'YTick',[-90 -88]);
    end
    hold on;
    [AX,H3,H4] = plotyy(position_INS(:,4),position_INS(:,1), position_INS(:,4),position_INS(:,2), 'plot');
    set(H3,'Color','b')
    set(H3,'LineStyle',':')
    set(H4,'Color','g')
    set(H4,'LineStyle',':')
    xlabel('Time (seconds)');
    legend('True Flightpath', 'INS Flightpath');
    title({'Aircraft Lattitude and Longitude versus Time'; titl}, 'FontSize', 16)
    axis(AX(1),[0 time 35 37])
    set(AX(1), 'YTick', [35 35.5 36 36.5 37]);
    if M=='R'
        axis(AX(2),[0 time -122 -120])
        set(AX(2),'YTick',[-122 -121.5 -121 -120.5 -120]);
    else
        axis(AX(2),[0 time -90 -88])
        set(AX(2),'YTick',[-90 -89.5 -89 -88.5 -88]);
    end

%Plot the true flightpath from a bird's eye view
    figure
    plot(position_true(:,2),position_true(:,1))
    xlabel('Longitude (degrees)'); ylabel('Lattitude (degrees)')
    if ((M=='R')|(M=='r'))
        xlim([-121.5,-120.5])
    else
        xlim([-89.5,-88.5])
    end
    ylim([35.5,36.5])
    title({'Flightpath'; titl}, 'FontSize', 16)

close(h)
```

Appendix B. Table of Results

The following tables present the results for all simulations discussed in this paper. The varying inputs for each section are highlighted in blue. Table 12 includes sensitivity analysis results for terrain, altitude, velocity, flight duration, INS drift, and position update rate. Table 13 includes results for GGI components and Table 14 for GGI and map noise, map resolution, and interpolation. Table 15 includes the practical scenario results (i.e. fighter, cargo, ISR, optimist, pessimist). Each table includes labels at the top.

The first 12 columns describe inputs. Column 1 indicates the terrain, where 'R' refers to the rough terrain in California, and 'S' refers to the smooth terrain in Tennessee, both detailed on page 52. Columns 2, 3, and 4 indicate the aircraft's altitude in meters, velocity in meters per second, and INS drift rate in meters per hour. Columns 6 and 7 indicate the rate in seconds that the GGI provides gravity gradient information and map matching solutions are attempted, respectively. Columns 8 and 9 indicate the noise in Eötvös introduced to the GGI and map. Columns 10, 11, and 12 indicate the Resolution Level (defined on page 58) of the source, simulated, and interpolated maps.

The remaining columns describe outputs (i.e. results). Column 13 presents the time in seconds for the computer to run the simulation. Columns 14, 15, and 16 present the mean of the RMS position errors, their standard deviation, and the CEP of the GAME solutions, all in meters. Columns 17, 18, and 19 present the same information for the GGI solutions. Columns 20 and 21 present the performance gains and break even points. Columns 22 and 23 present the number of failed and attempted matches. Only Table 13 includes column 24, which lists the components of the gravity gradient tensor included in the simulation. The simulations in the other tables include all five components.

Table 12: Sensitivity Analysis Results
(Terrain, Altitude, Velocity, Duration, INS Drift, Update Rate)

Terrain	Altitude (m)	Velocity (m/s)	Flight Duration (hr)	INS Drift (m/hr)	GGI Update (s)	GAME Update (s)	GGI Noise (Eo)	Map Noise (Eo)	Source (AGED)	Sim (AGED)	Interp (AGED)	Processor Time (s)	GAME RMS (m)	GAME std dev (m)	GAME CEP (m)	GGI RMS (m)	GGI std dev (m)	GGI CEP (m)	Performance Gain	BEP (min)	Failed Matches	Attempted Matches
TERRAIN																						
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	80.1	128.9	25.4	128.3	174.3	126.2	141.4	34.7	3.8	0	8000
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	132.7	407.3	98.8	424.5	543.4	597.8	378.3	10.5	12.7	0	8000
ALTITUDE																						
R	1000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	102.7	127.5	22.5	122.4	121.8	54.7	122.0	36.3	3.7	528	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	88.3	128.3	23.6	129.4	174.1	125.8	140.9	34.3	3.9	0	8000
R	10000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	90.0	192.9	37.0	193.0	355.3	277.9	275.0	23.0	5.8	0	8000
R	15000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	107.9	246.5	58.0	243.0	547.0	459.4	412.0	18.3	7.3	0	8000
R	20000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	132.6	304.8	70.1	313.0	716.2	571.6	558.9	14.2	9.4	0	8000
R	25000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	167.5	403.1	96.3	421.6	892.7	670.6	719.8	10.5	12.6	0	8000
R	30000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	200.6	499.8	131.0	525.5	1124.7	804.0	921.8	8.5	15.8	0	8000
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	150.4	417.7	108.6	439.6	537.3	578.9	378.9	10.1	13.2	0	8000
S	10000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	307.0	809.1	277.4	905.3	1270.0	1155.6	972.8	4.9	27.2	0	8000
S	15000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	394.2	922.9	344.7	1029.8	1673.0	1414.7	1315.3	4.3	30.9	0	8000
S	20000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	460.5	1096.8	465.3	1222.0	1997.5	1650.8	1613.3	3.6	36.7	0	8000
S	25000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	519.3	1194.7	523.2	1337.5	2306.4	1851.6	1862.9	3.3	40.1	0	8000
S	30000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	583.1	1315.9	617.1	1448.9	2598.7	2108.2	2097.2	3.1	43.5	0	8000
VELOCITY																						
R	5000	25	13.3333	2000	1	1	0.1	0.01	1	1	1	1093.8	138.4	33.4	135.2	183.0	161.7	143.1	197.3	4.1	0	48000
R	5000	50	6.66667	2000	1	1	0.1	0.01	1	1	1	381.1	135.1	30.0	133.0	179.0	143.1	141.8	100.3	4.0	0	24000
R	5000	100	3.33333	2000	1	1	0.1	0.01	1	1	1	136.9	133.7	26.7	131.9	179.1	142.8	142.6	50.5	4.0	0	12000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	92.8	130.9	25.5	129.1	177.9	142.3	141.3	34.4	3.9	0	8000
R	5000	200	1.66667	2000	1	1	0.1	0.01	1	1	1	74.7	131.5	26.8	129.3	177.8	142.6	140.9	25.8	3.9	0	6000
R	5000	250	1.33333	2000	1	1	0.1	0.01	1	1	1	63.2	131.0	25.6	129.4	177.2	141.2	140.8	20.6	3.9	0	4800
R	5000	300	1.11111	2000	1	1	0.1	0.01	1	1	1	56.1	129.9	26.0	128.4	176.7	140.9	140.6	17.3	3.9	0	4000
R	5000	350	0.952381	2000	1	1	0.1	0.01	1	1	1	51.9	132.1	25.7	130.9	177.9	141.5	141.6	14.6	3.9	0	3429
R	5000	400	0.833333	2000	1	1	0.1	0.01	1	1	1	34.6	124.5	25.7	124.4	170.7	116.4	140.6	13.4	3.7	0	3000
R	5000	450	0.740741	2000	1	1	0.1	0.01	1	1	1	32.4	130.3	28.8	131.6	172.7	113.0	144.9	11.3	3.9	0	2667
R	5000	500	0.666667	2000	1	1	0.1	0.01	1	1	1	31.4	124.1	25.9	125.3	169.6	114.9	140.3	10.6	3.8	0	2400
R	5000	550	0.606061	2000	1	1	0.1	0.01	1	1	1	30.9	123.9	23.6	126.1	171.0	116.6	140.8	9.6	3.8	0	2182
R	5000	600	0.555556	2000	1	1	0.1	0.01	1	1	1	29.9	125.5	23.0	128.5	171.3	117.0	140.7	8.6	3.9	0	2000
R	5000	650	0.512821	2000	1	1	0.1	0.01	1	1	1	29.7	122.6	25.8	124.9	168.5	118.3	138.1	8.2	3.7	0	1846
R	5000	700	0.47619	2000	1	1	0.1	0.01	1	1	1	29.0	131.4	26.7	134.5	175.3	121.4	142.6	7.1	4.0	0	1714
R	5000	750	0.444444	2000	1	1	0.1	0.01	1	1	1	20.9	130.2	31.0	136.7	164.4	103.9	140.6	6.5	4.1	0	1600
R	5000	800	0.416667	2000	1	1	0.1	0.01	1	1	1	20.6	131.7	31.0	137.4	170.8	107.3	144.4	6.1	4.1	0	1500
R	5000	850	0.392157	2000	1	1	0.1	0.01	1	1	1	20.8	117.3	24.9	121.4	163.0	99.1	139.3	6.5	3.6	0	1412
R	5000	900	0.37037	2000	1	1	0.1	0.01	1	1	1	20.3	128.1	30.2	130.8	173.0	103.4	147.0	5.7	3.9	0	1333
R	5000	950	0.350877	2000	1	1	0.1	0.01	1	1	1	20.4	118.0	31.2	121.9	169.4	108.3	142.4	5.8	3.7	0	1263
R	5000	1000	0.333333	2000	1	1	0.1	0.01	1	1	1	20.5	115.3	30.5	122.5	169.6	104.8	142.4	5.4	3.7	0	1200
R	5000	1050	0.31746	2000	1	1	0.1	0.01	1	1	1	16.4	121.4	33.4	132.6	166.8	99.5	141.4	4.8	4.0	0	1143
R	5000	1100	0.30303	2000	1	1	0.1	0.01	1	1	1	16.1	122.8	33.1	128.3	169.1	101.4	143.4	4.7	3.8	0	1091
R	5000	1150	0.289855	2000	1	1	0.1	0.01	1	1	1	16.2	115.0	31.0	121.7	168.9	101.4	144.4	4.8	3.7	0	1043
R	5000	1200	0.277778	2000	1	1	0.1	0.01	1	1	1	16.4	131.6	35.0	141.8	169.8	100.7	144.2	3.9	4.3	0	1000
R	5000	1250	0.266667	2000	1	1	0.1	0.01	1	1	1	16.6	119.0	34.3	126.7	164.8	97.7	142.4	4.2	3.8	0	960
FLIGHT DURATION																						
R	5000	150	0.5	2000	1	1	0.1	0.01	1	1	1	22.5	131.2	35.5	133.2	175.9	115.2	146.3	7.5	4.0	0	1800
R	5000	150	1	2000	1	1	0.1	0.01	1	1	1	42.5	131.6	34.0	130.4	164.4	109.1	136.4	15.3	3.9	0	3600
R	5000	150	2	2000	1	1	0.1	0.01	1	1	1	89.7	126.0	24.6	126.7	171.9	122.1	140.8	31.6	3.8	0	7200
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	78.5	128.9	25.4	128.3	174.3	126.2	141.4	34.7	3.8	0	8000
R	5000	150	4	2000	1	1	0.1	0.01	1	1	1	212.6	128.1	23.4	127.0	175.2	128.0	141.2	63.0	3.8	0	14400
R	5000	150	8	2000	1	1	0.1	0.01	1	1	1	546.1	130.1	23.0	130.4	175.5	134.1	141.2	122.7	3.9	0	28800
R	5000	150	16	2000	1	1	0.1	0.01	1	1	1	1545.2	130.7	23.7	130.3	174.4	138.5	141.1	245.5	3.9	0	57600
R	5000	150	24	2000	1	1	0.1	0.01	1	1	1	2935.2	130.0	26.0	130.0	174.8	157.4	140.9	369.1	3.9	0	86400
R	5000	150	32	2000	1	1	0.1	0.01	1	1	1	4949.7	129.5	25.9	129.2	175.7	179.4	139.6	495.4	3.9	0	115200
INS DRIFT																						
R	5000	150	2.22222	0.02	1	1	0.1	0.01	1	1	1	143.0	0.0	0.0	0.0	162.4	91.6	140.9	2.4	56.0	0	8000
R	5000	150	2.22222	0.2	1	1	0.1	0.01	1	1	1	141.7	0.2	0.1	0.2	163.4	93.0	141.1	2.4	56.1	0	8000
R	5000	150	2.22222	2	1	1	0.1	0.01	1	1	1	148.7	1.9	1.1	1.9	161.7	92.2	140.8	2.3	57.0	0	8000
R	5000	150	2.22222	20	1	1	0.1	0.01	1	1	1	153.7	24.4	15.9	23.5	160.5	91.0	140.6	1.9	70.4	0	8000
R	5000	150	2.22222	200	1	1	0.1	0.01	1	1	1	142.4	111.0	33.9	123.5	159.0	94.5	138.3	3.6	37.1	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	111.1	130.1	26.2	128.8	175.0	123.2	141.2	34.5	3.9	0	8000
R	5000	150	2.22222	20000	1	1	0.1	0.01	1	1	1	219.2	141.0	63.6	131.2	185.4	192.8	141.3	338.7	0.4	0	8000
R	5000	150	2.22222	200000	1	1	0.1	0.01	1	1	1	2549.8	170.6	159.7	138.8	184.8	198.4	141.2	3201.7	0.0	0	8000
POSITION UPDATE RATE																						
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	88.8	128.0	24.0	127.2	174.5	132.1	140.7	34.9	3.8	0	8000
R	5000	150	2.22222	2000	1	15	0.1	0.01	1	1	1	37.1	192.1	45.8	201.5	168.4	120.6	139.0	22.1	6.0	0	533
R	5000	150	2.22222	2000	1	30	0.1	0.01	1	1	1	35.9	244.3	55.3	245.4	174.9	123.1	140.8	18.1	7.4	0	266
R	5000	150	2.22222	2000	1	45	0.1	0.01	1	1	1	41.0	263.5	68.9	265.7	171.1	124.3	137.9	16.7	8.0	0	177
R	5000	150	2.22222	2000	1	60	0.1	0.01	1	1	1	32.9	316.4	96.1	337.3	211.7	273.3	135.5	13.2	10.1	0	133
R	5000	150	2.22222	2000	1	900	0.1	0.01	1	1	1	31.7	963.6	353.8	1064.8	245.3	200.5	211.4	4.2	31.9	0	8
R	5000	150	2.22222	2000	1	1800	0.1	0.01	1	1	1	33.6	1074.9	432.4	1165.2	191.2	103.4	177.2	3.8	35.0	0	4
R	5000	150	2.22222	2000	1	2700	0.1	0.01	1	1	1	39.1	1323.9	614.7	1364.1	199.1	99.5	190.2	3.3	40.9	0	2
R	5000	150	2.22222	2000	1	3600	0.1	0.01	1	1	1	44.6	1210.1	569.7	1267.1	222.9	113.7	205.0	3.5	38.0	0	2
R	5000	150	2.22222	2000	1	7999	0.1	0.01	1	1	1	31.8	1859.7	1074.3	1865.9	83.0	0.0	83.0	2.4	56.0	0	1

Table 13: Sensitivity Analysis Results (GGI Components)

Terrain	Altitude (m)	Velocity (m/s)	Flight Duration (hr)	INS Drift (m/hr)	GGI Update (s)	GAME Update (s)	GGI Noise (Eo)	Map Noise (Eo)	Source (AGED)	Sim (AGED)	Interp (AGED)	Processor Time (s)	GAME RMS (m)	GAME std dev (m)	GAME CEP (m)	GGI RMS (m)	GGI std dev (m)	GGI CEP (m)	Performance Gain	BEP (min)	Failed Matches	Attempted Matches	GGI Components
COMPONENTS																							
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	93.4	868.6	398.3	900.9	1063.1	840.9	832.2	4.9	27.0	0	8000	Txx
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	96.8	612.3	301.3	578.0	1028.9	830.0	803.5	7.7	17.3	0	8000	Txy
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	92.0	916.9	294.2	939.4	993.9	785.8	785.5	4.7	28.2	0	8000	Txz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	91.9	426.9	204.7	387.9	817.9	644.0	637.9	11.5	11.6	0	8000	Tyz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	88.7	311.1	131.8	296.0	683.3	509.5	564.3	15.0	8.9	0	8000	Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	419.1	1693.3	1008.6	1674.2	2687.6	2452.5	1963.5	2.7	50.2	0	8000	Txx
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	411.9	1356.1	683.3	1487.8	2635.9	2469.4	1849.3	3.0	44.6	0	8000	Txy
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	360.2	1529.1	805.9	1491.4	2436.4	2316.7	1694.1	3.0	44.7	0	8000	Txz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	315.1	1105.2	518.5	1263.1	2002.7	2061.1	1265.4	3.5	37.9	0	8000	Tyz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	284.8	1337.6	757.6	1284.5	1968.7	1998.9	1273.6	3.5	38.5	0	8000	Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	86.0	142.8	56.5	139.5	404.4	369.4	273.2	31.9	4.2	0	8000	Txx/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	85.6	160.9	78.8	145.7	404.8	381.6	263.7	30.5	4.4	0	8000	Txy/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	91.8	174.5	76.9	147.9	348.8	340.8	224.2	30.1	4.4	0	8000	Txz/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	97.2	162.0	56.7	154.1	336.5	340.5	210.9	28.8	4.6	0	8000	Tyz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	207.8	1056.8	511.5	1014.6	1460.1	1536.6	895.1	4.4	30.4	0	8000	Txx/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	210.3	860.3	421.4	939.7	1361.8	1496.2	819.7	4.7	28.2	0	8000	Txy/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	198.1	621.4	213.3	635.1	1203.9	1255.3	747.0	7.0	19.1	0	8000	Txz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	195.0	635.0	219.4	669.8	948.8	1119.3	558.7	6.6	20.1	0	8000	Tyz/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	86.9	136.6	45.5	130.4	281.8	265.8	193.1	34.1	3.9	0	8000	Txx/Txy/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	85.3	143.0	44.4	132.5	261.8	252.9	177.4	33.5	4.0	0	8000	Txx/Txz/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	91.7	139.0	31.1	137.3	249.5	236.7	170.4	32.4	4.1	0	8000	Txx/Tyz/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	86.7	143.7	47.0	133.8	258.1	251.9	176.0	33.2	4.0	0	8000	Txy/Txz/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	87.4	142.1	45.9	135.0	252.3	247.5	170.7	32.9	4.1	0	8000	Txy/Tyz/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	86.7	139.5	31.0	137.9	223.4	209.9	154.6	32.2	4.1	0	8000	Txz/Tyz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	181.9	676.2	246.9	740.8	1059.7	1166.4	657.7	6.0	22.2	0	8000	Txx/Txy/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	174.9	526.9	163.4	531.7	974.2	1045.9	622.8	8.4	16.0	0	8000	Txx/Txz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	253.8	552.4	169.0	574.4	784.3	904.7	494.2	7.7	17.2	0	8000	Txx/Tyz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	175.6	522.2	164.0	539.5	905.0	1014.3	570.8	8.2	16.2	0	8000	Txy/Txz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	164.0	545.4	179.4	558.3	733.1	915.2	447.2	8.0	16.7	0	8000	Txy/Tyz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	164.5	476.9	142.0	480.1	749.5	865.4	469.2	9.3	14.4	0	8000	Txz/Tyz/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	85.9	137.4	39.3	130.6	199.1	166.9	151.0	34.0	3.9	0	8000	Txx/Txy/Txz/Tyz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	86.5	131.1	34.8	126.1	214.2	191.1	155.6	35.2	3.8	0	8000	Txx/Txy/Txz/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	85.7	136.0	30.6	133.5	208.6	179.5	153.0	33.3	4.0	0	8000	Txx/Txy/Tyz/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	86.5	133.4	31.2	129.3	196.8	163.7	145.2	34.4	3.9	0	8000	Txx/Txz/Tyz/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	86.7	134.7	34.2	130.8	198.0	168.5	147.8	34.0	3.9	0	8000	Txy/Txz/Tyz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	172.3	474.7	127.4	491.0	742.4	796.6	508.3	9.1	14.7	0	8000	Txx/Txy/Txz/Tyz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	166.4	480.2	147.8	496.1	770.4	858.1	507.2	9.0	14.9	0	8000	Txx/Txy/Txz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	157.4	480.4	133.3	504.8	646.1	766.4	417.7	8.8	15.1	0	8000	Txx/Txy/Tyz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	151.2	422.5	108.9	435.6	650.1	736.9	418.9	10.2	13.1	0	8000	Txx/Txz/Tyz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	152.4	449.6	120.2	469.1	611.8	713.1	408.0	9.5	14.1	0	8000	Txy/Txz/Tyz/Tzz
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	86.1	130.9	23.2	132.3	175.9	130.1	141.2	33.6	4.0	0	8000	Txx/Txy/Txz/Tyz/Tzz
S	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	151.4	412.1	106.1	427.9	549.1	616.1	388.9	10.4	12.8	0	8000	Txx/Txy/Txz/Tyz/Tzz

Table 14: Sensitivity Analysis Results (GGI/Map Noise, Resolution, Interpolation)

Terrain	Altitude (m)	Velocity (m/s)	Flight Duration (hr)	INS Drift (m/hr)	GGI Update (s)	GAME Update (s)	GGI Noise (Eo)	Map Noise (Eo)	Source (AGED)	Sim (AGED)	Interp (AGED)	Processor Time (s)	GAME RMS (m)	GAME std dev (m)	GAME CEP (m)	GGI RMS (m)	GGI std dev (m)	GGI CEP (m)	Performance Gain	BEP (min)	Failed Matches	Attempted Matches
GGI NOISE																						
R	5000	150	2.22222	2000	1	1	1.00E-06	0.01	1	1	1	102.6	109.6	16.0	115.4	109.4	42.7	112.6	38.5	3.5	0	8000
R	5000	150	2.22222	2000	1	1	1.00E-05	0.01	1	1	1	103.2	109.6	15.1	114.1	109.3	42.6	112.3	38.9	3.4	0	8000
R	5000	150	2.22222	2000	1	1	0.0001	0.01	1	1	1	108.6	109.2	15.3	113.9	109.4	42.2	112.6	39.0	3.4	0	8000
R	5000	150	2.22222	2000	1	1	0.001	0.01	1	1	1	114.2	109.9	15.0	115.2	109.3	42.3	112.4	38.6	3.5	0	8000
R	5000	150	2.22222	2000	1	1	0.01	0.01	1	1	1	119.4	111.1	15.2	114.9	110.9	43.5	113.1	38.7	3.4	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	107.9	123.2	25.8	123.6	174.9	127.8	141.1	35.9	3.7	0	8000
R	5000	150	2.22222	2000	1	1	1	0.01	1	1	1	160.9	380.1	99.1	388.1	1064.3	916.6	775.3	11.5	11.6	0	8000
R	5000	150	2.22222	2000	1	1	5	0.01	1	1	1	359.0	1085.5	431.7	1183.1	3292.4	2437.5	2668.4	3.8	35.5	0	8000
MAP NOISE																						
R	5000	150	2.22222	2000	1	1	0.1	1.00E-06	1	1	1	90.1	126.5	24.2	126.8	170.4	122.2	140.6	35.1	3.8	0	8000
R	5000	150	2.22222	2000	1	1	0.1	1.00E-05	1	1	1	88.3	122.0	23.0	123.8	171.0	121.8	140.6	35.9	3.7	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.0001	1	1	1	92.9	128.6	24.1	127.9	172.3	122.8	140.7	34.8	3.8	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.001	1	1	1	98.6	125.5	24.5	125.5	173.3	125.8	140.9	35.4	3.8	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	104.3	122.2	22.3	122.9	172.4	125.3	140.6	36.2	3.7	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.1	1	1	1	107.2	151.6	48.8	141.3	222.4	177.5	168.2	31.5	4.2	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.5	1	1	1	101.7	324.1	168.3	308.4	577.1	501.0	421.5	14.4	9.3	0	8000
R	5000	150	2.22222	2000	1	1	0.1	1	1	1	1	155.6	417.0	189.4	408.0	949.7	785.9	715.6	10.9	12.2	0	8000
R	5000	150	2.22222	2000	1	1	0.1	5	1	1	1	466.7	1509.6	739.2	1560.2	3038.6	2516.1	2322.0	2.8	46.8	0	8000
R	5000	150	2.22222	2000	1	1	0.1	10	1	1	1	724.7	1848.6	1029.1	1889.2	4150.3	3360.2	3272.2	2.4	56.7	0	8000
MAP RESOLUTION																						
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	-5	1	242.5	254.2	134.4	242.2	296.1	252.1	251.7	18.3	7.3	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	-4	1	105.2	162.1	66.1	153.5	207.5	180.1	172.2	29.0	4.6	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	-3	1	84.1	131.8	39.4	131.2	189.8	155.7	148.2	33.9	3.9	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	-2	1	92.7	129.7	27.5	128.0	175.5	127.6	141.5	34.7	3.8	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	-1	1	104.4	127.1	25.2	126.7	171.0	123.1	140.7	35.1	3.8	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	0	1	108.1	127.5	24.5	126.4	174.2	133.0	140.6	35.1	3.8	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	116.2	129.1	24.4	128.9	174.4	127.9	141.0	34.5	3.9	0	8000
MAP INTERPOLATION																						
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	1	88.2	126.2	26.2	125.6	175.2	130.0	141.1	35.4	3.8	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	2	98.3	125.6	24.5	126.6	170.1	122.0	138.7	35.1	3.8	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	3	180.6	124.0	26.2	123.4	170.3	126.3	137.1	36.0	3.7	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	4	634.0	130.3	28.9	128.0	169.8	121.2	138.7	34.7	3.8	0	8000
R	5000	150	2.22222	2000	1	1	0.1	0.01	1	1	5	2524.4	129.1	30.4	125.6	171.1	123.0	138.8	35.4	3.8	0	8000

Table 15: Practical Scenario Results
(Fighter, Cargo, ISR, Optimist, Pessimist)

Terrain	Altitude (m)	Velocity (m/s)	Flight Duration (hr)	INS Drift (m/hr)	GGI Update (s)	GAME Update (s)	GGI Noise (Eo)	Map Noise (Eo)	Source (AGED)	Sim (AGED)	Interp (AGED)	Processor Time (s)	GAME RMS (m)	GAME std dev (m)	GAME CEP (m)	GGI RMS (m)	GGI std dev (m)	GGI CEP (m)	Performance Gain	BEP (min)	Failed Matches	Attempted Matches
FIGHTER MISSION																						
S	5000	400	1.5	20	1	1	0.01	0.001	1	1	3	105.4	19.2	13.5	17.6	105.9	52.8	100.6	1.7	52.9	0	5400
S	5000	400	1.5	20	1	1	0.1	0.01	1	1	3	108.2	15.9	10.7	14.7	142.1	84.3	129.6	2.0	44.1	0	5400
S	5000	400	1.5	20	1	1	1	0.1	1	1	3	112.5	16.3	11.1	14.9	151.8	81.1	140.9	2.0	44.6	0	5400
S	5000	400	1.5	200	1	1	0.01	0.001	1	1	3	94.8	137.2	47.7	149.6	127.9	76.0	123.4	2.0	44.9	0	5400
S	5000	400	1.5	200	1	1	0.1	0.01	1	1	3	105.3	142.7	70.9	148.8	294.8	259.2	229.2	2.0	44.6	0	5400
S	5000	400	1.5	200	1	1	1	0.1	1	1	3	112.1	154.5	79.1	160.6	388.7	328.9	286.4	1.9	48.2	0	5400
S	5000	400	1.5	2000	1	1	0.01	0.001	1	1	3	155.1	130.2	25.5	133.3	123.0	116.4	100.3	22.5	4.0	0	5400
S	5000	400	1.5	2000	1	1	0.1	0.01	1	1	3	765.8	461.8	130.7	510.0	674.6	728.4	442.1	5.9	15.3	0	5400
S	5000	400	1.5	2000	1	1	1	0.1	1	1	3	227.4	1066.6	595.2	1058.3	2712.3	2444.5	1980.1	2.8	31.7	0	5400
CARGO MISSION																						
R	10000	250	2	2000	1	1	0.01	0.001	1	1	1	139.9	122.5	19.1	118.9	123.7	48.3	118.8	33.6	3.6	0	7200
R	10000	250	2	2000	1	1	0.1	0.01	1	1	1	151.0	175.8	37.0	179.2	347.6	268.9	271.4	22.3	5.4	0	7200
R	10000	250	2	2000	1	1	1	0.1	1	1	1	410.7	727.8	231.2	789.5	2262.9	1742.7	1777.1	5.1	23.7	0	7200
R	10000	250	4	2000	1	1	0.01	0.001	1	1	1	396.6	124.3	17.9	120.5	124.5	48.7	119.4	66.4	3.6	0	14400
R	10000	250	4	2000	1	1	0.1	0.01	1	1	1	340.4	181.0	32.3	183.4	360.9	299.1	274.7	43.6	5.5	0	14400
R	10000	250	4	2000	1	1	1	0.1	1	1	1	1588.5	721.3	169.8	744.6	2772.7	2285.9	2086.5	10.7	22.3	0	14400
R	10000	250	8	2000	1	1	0.01	0.001	1	1	1	889.2	126.5	17.4	121.8	126.4	49.5	120.4	131.4	3.7	0	28800
R	10000	250	8	2000	1	1	0.1	0.01	1	1	1	978.0	185.7	32.5	185.7	371.8	331.0	277.1	86.1	5.6	0	28800
R	10000	250	8	2000	1	1	1	0.1	1	1	1	6476.5	733.6	127.2	753.6	3259.1	2952.0	2293.1	21.2	22.6	0	28800
R	10000	250	16	2000	1	1	0.01	0.001	1	1	1	1347.1	125.1	18.3	120.7	126.4	50.3	120.1	265.2	3.6	0	57600
R	10000	250	16	2000	1	1	0.1	0.01	1	1	1	1608.7	177.7	33.0	176.8	375.7	365.2	274.3	181.0	5.3	0	57600
R	10000	250	16	2000	1	1	1	0.1	1	1	1	14526.8	693.4	133.2	714.4	3633.7	3605.0	2391.9	44.8	21.4	0	57600
ISR MISSION																						
R	5000	150	24	200	1	1	0.01	0.001	1	1	1	2122.8	115.4	12.4	115.8	113.3	41.0	114.0	41.4	34.8	332	86400
R	5000	150	24	200	1	1	0.1	0.01	1	1	1	930.3	124.8	14.2	126.2	170.4	120.8	140.8	38.0	37.9	0	86400
R	5000	150	24	200	1	1	1	0.1	1	1	1	1284.3	338.6	62.9	353.5	1000.7	856.6	736.9	13.6	106.1	0	86400
R	15000	150	24	200	1	1	0.01	0.001	1	1	1	3289.8	124.8	10.8	125.4	132.9	60.9	120.5	38.3	37.6	0	86400
R	15000	150	24	200	1	1	0.1	0.01	1	1	1	2171.3	218.1	33.6	221.2	531.8	442.3	398.8	21.7	66.4	0	86400
R	15000	150	24	200	1	1	1	0.1	1	1	1	10486.6	1023.1	345.5	1214.8	3321.2	2483.1	2681.2	4.0	364.5	0	86400
R	25000	150	24	200	1	1	0.01	0.001	1	1	1	3078.6	130.9	15.4	131.5	146.6	78.3	132.0	36.5	39.4	0	86400
R	25000	150	24	200	1	1	0.1	0.01	1	1	1	3416.2	390.9	77.1	415.7	873.9	661.9	698.3	11.5	124.7	0	86400
R	25000	150	24	200	1	1	1	0.1	1	1	1	13697.5	1556.3	720.4	1747.7	4759.0	3402.9	4079.2	2.7	524.3	0	86400
OPTIMIST																						
R	5000	150	2	20	1	1	0.1	0.01	1	1	1	126.4	21.5	13.9	21.1	161.6	91.6	141.2	1.9	63.4	0	7200
R	5000	150	2	200	1	1	0.1	0.01	1	1	1	118.9	107.7	36.1	122.0	156.9	90.9	139.0	3.3	36.6	0	7200
R	5000	150	2	2000	1	1	0.1	0.01	1	1	1	86.6	126.8	26.6	127.3	174.1	122.0	141.3	31.4	3.8	0	7200
R	5000	150	4	20	1	1	0.1	0.01	1	1	1	357.0	48.0	29.4	48.1	160.1	89.7	140.7	1.7	144.2	0	14400
R	5000	150	4	200	1	1	0.1	0.01	1	1	1	351.9	118.4	28.0	125.7	165.9	103.0	140.9	6.4	37.7	0	14400
R	5000	150	4	2000	1	1	0.1	0.01	1	1	1	360.7	128.6	24.5	127.2	175.6	130.8	141.2	62.9	3.8	0	14400
R	5000	150	8	20	1	1	0.1	0.01	1	1	1	508.3	79.7	39.0	91.9	156.7	88.1	138.9	1.7	275.8	0	28800
R	5000	150	8	200	1	1	0.1	0.01	1	1	1	363.1	120.4	20.1	125.1	168.3	109.1	141.0	12.8	37.5	0	28800
R	5000	150	8	2000	1	1	0.1	0.01	1	1	1	446.2	128.6	23.1	127.6	174.3	129.6	141.0	125.4	3.8	0	28800
R	5000	150	16	20	1	1	0.1	0.01	1	1	1	983.5	105.7	37.8	124.7	155.7	88.8	137.4	2.6	374.0	0	57600
R	5000	150	16	200	1	1	0.1	0.01	1	1	1	675.8	124.5	16.0	127.2	168.9	115.2	140.7	25.2	38.2	0	57600
R	5000	150	16	2000	1	1	0.1	0.01	1	1	1	1115.0	131.8	24.0	131.6	174.6	135.9	141.7	243.2	3.9	0	57600
PESSIMIST																						
S	15000	150	2	20	1	1	0.1	0.01	1	1	1	142.8	14.8	8.7	14.6	262.8	165.9	229.3	2.7	43.8	0	7200
S	15000	150	2	200	1	1	0.1	0.01	1	1	1	109.8	161.8	92.5	164.9	517.8	402.4	430.3	2.4	49.5	0	7200
S	15000	150	2	2000	1	1	0.1	0.01	1	1	1	391.4	1048.3	457.0	1135.4	2217.0	1876.7	1714.1	3.5	34.1	0	7200
S	15000	150	4	20	1	1	0.1	0.01	1	1	1	252.3	30.8	18.5	30.2	265.0	173.6	223.9	2.7	90.5	0	14400
S	15000	150	4	200	1	1	0.1	0.01	1	1	1	181.2	311.8	173.0	319.2	899.9	723.1	712.3	2.5	95.8	0	14400
S	15000	150	4	2000	1	1	0.1	0.01	1	1	1	1630.2	1363.7	446.2	1542.6	2688.4	2341.9	2060.5	5.2	46.3	0	14400
S	15000	150	8	20	1	1	0.1	0.01	1	1	1	711.5	65.0	39.3	63.9	293.4	204.7	233.0	2.5	191.8	0	28800
S	15000	150	8	200	1	1	0.1	0.01	1	1	1	964.3	569.7	300.8	595.0	1444.8	1196.9	1137.9	2.7	178.5	0	28800
S	15000	150	8	2000	1	1	0.1	0.01	1	1	1	1620.3	420.1	1350.3	3001.1	2687.3	12111.4	2255.1	9.3	51.4	0	28800
S	15000	150	16	20	1	1	0.1	0.01	1	1	1	867.0	130.8	75.9	132.9	436.8	336.5	365.9	2.4	398.7	0	57600
S	15000	150	16	200	1	1	0.1	0.01	1	1	1	2362.0	930.2	428.5	1033.9	2002.6	1695.3	1545.1	3.1	310.2	0	57600
S	15000	150	16	2000	1	1	0.1	0.01	1	1	1	31191.3	1961.4	505.5	1955.5	3385.4	3430.3	2382.2	16.4	58.7	0	57600

Bibliography

[1] Schwartz, N. A., "Keynote Address at the Air, Space, and Cyberspace Power in the 21st Century Conference," *Air, Space, and Cyberspace Power in the 21st Century Conference*, Washington, DC, 20 January 2010, URL: http://www.af.mil/shared/media/document/AFD-100121-002.pdf. [cited 19 February 2010].

[2] Jekeli, C., *Inertial Navigation Systems with Geodetic Applications*, Walter de Gruyter, New York, 2001.

[3] Grewal, M. S., Weill, L. R. Weill, and Andrews, A. P., *Global Positioning Systems, Inertial Navigation, and Integration*, 2nd ed., John Wiley & Sons, Inc., Hoboken, NJ, 2007.

[4] Clausing, D. J., *Aviator's Guide to Navigation*, 3rd ed., McGraw-Hill, New York, 1997.

[5] MacKenzie, D., *Inventing Accuracy: A Historical Sociology of Nuclear Missile Guidance*, Massachusetts Institute of Technology, 1993.

[6] Bonner, M., "Inertial Navigation Systems," Powerpoint Presentation, 412th Test Wing, Edwards Air Force Base, CA.

[7] Titterton, D. H., and Weston, J. L., *Strapdown Inertial Navigation Technology*, 2nd ed, AIAA, Reston, VA, 2004.

[8] Biezad, D. J., *Integrated Navigation and Guidance Systems*, AIAA, Reston, VA, 1999.

[9] Grewal, M. S., *Kalman Filtering: Theory and Practice Using MATLAB*, John Wiley & Sons, Inc., Hoboken, NJ, 2008.

[10] Chatfield, A. B., *Fundamentals of High Accuracy Inertial Navigation*, AIAA, Reston, VA, 1997.

[11] Hofmann-Wellenhof, B., and Moritz, H., *Physical Geodesy*, 2nd ed., Springer-Verlag Wien, New York, 2006.

[12] Richeson, J. A., "Gravity Gradiometry Aided Inertial Navigation within Non-GNSS Environments," Ph.D. Dissertation, University of Maryland, College Park, MD, 2008.

[13] Mickus, K. L., and Hinojosa, J. H., "The complete gravity gradient tensor derived from the vertical component of gravity: a Fourier transform technique," *Journal of Applied Geophysics*, Vol. 46, 2001, pp. 159-174.

[14] Jekeli, C., "Precision Free-Inertial Navigation with Gravity Compensation by and Onboard Gradiometer," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 3, May-June 2006, pp. 704-713.

[15] Bell, Robin E. "Gravity gradiometry," *Scientific American*, Vol. 278, No. 6: 74 (June 1998).

[16] "Loránd Eötvös Virtual Museum," Nemzeti Kultúrális Alap and the Eötvös Loránd Geophysical Institute of Hungary. URL: http://www.elgi.hu/museum/tinga_.htm [cited 14 September 2009].

[17] Torge, W., *Geodesy*, Walter de Gruyter, New York, 2001.

[18] Nagy, D., Papp, G., Benedek, J., "The gravitational potential and its derivatives for the prism," *Journal of Geodesy*, Vol. 74, 2000, pp. 552-560.

[19] Rogers, M. M., "An Investigation into the Feasibility of Using a Modern Gravity Gradient Instrument for Passive Aircraft Navigation and Terrain Avoidance," Master's Thesis, Department of Aeronautics and Astronautics, Air Force Institute of Technology, Wright-Patterson AFB, OH, March 2009.

[20] Veryaskin, A., and McRae, W., "On combined gravity gradient components modeling for applied geophysics," *Journal of Geophysics and Engineering*, Vol. 5, 2008, pp. 348-356.

[21] Greenfield, J. S. "Matching GPS Observations to Locations on a Digital Map," Department of Civil and Environmental Engineering, New Jersey Institute of Technology, Newark, NJ, 2002, URL: http://www.njtide.org/reports/TRB2002-greenfeld.pdf [cited 15 September 2009].

[22] Gallagher, B., "The State of the Art in Graph-Based Pattern Matching," Lawrence Livermore National Laboratory, UCRL-TR-220300, 31 March 2006.

[23] Dedeoglu, G., and Sukhatme, G. S., "Landmark-based Matching Algorithm for Cooperative Mapping by Autonomous Robots," *Proceedings of the 5th International Symposium on Distributed Autonomous Robot Systems*, Knoxville, TN, 2000, pp. 251-260.

[24] Pink, O. "Visual map matching and localization using a global feature map," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Anchorage, AK, June 2008.

[25] Efrat, A., Indyk, P., and Venkatasubramanian, S., "Pattern Matching for Sets of Segments," Computer Science Department, Stanford University, February 2008.

[26] Hagen, O. K., "Terrain Navigation Principals and Application," September 2005, URL: http://www.navlab.net/Publications/Terrain_Navigation_Principles_Application.pdf, [cited 17 September 2009].

[27] Hagen, O. K., "TerrLab – a generic simulation and post-processing tool for terrain reference navigation," September 2006, Institute of Electrical and Electronics Engineers, Inc., URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4098989&isnumber=4098825?tag=1 [cited 2 October 2009].

[28] Archibald, J. B., "Two novel approaches to navigation using the Earth's gravity and magnetic fields," Ph.D. Dissertation, State University of New York at Buffalo, 1993.

[29] Di Massa, D. E., "Terrain-Relative Navigation for Autonomous Underwater Vehicles," Ph.D. Dissertation, Department of Ocean Engineering, Massachusetts Institute of Technology, May 1997.

[30] Dumrongchai, P. "Small Anomalous Detection from Airborne Gradiometry," Ph.D. Dissertation, The Ohio State University, 2007.

[31] Storms, W. F., "Magnetic Field Aided Indoor Navigation," Master's Thesis, Department of Aeronautics and Astronautics, Air Force Institute of Technology, Wright-Patterson AFB, OH, March 2009.

[32] Nygren, I., "Robust and Efficient Terrain Navigation of Underwater Vehicles," *IEEE/ION Position, Location and Navigation Symposium*, Piscataway, NJ, 2008, pp. 923-932..

[33] Nygren, I. "Terrain Navigation for Underwater Vehicles Using the Correlator Method," *IEEE Journal of Oceanic Engineering*, Vol. 29, July 2004, pp. 906-915.

[34] Bergman, N. "Bayesian Inference in Terrain Navigation," Thesis No. 649, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1997.

[35] "Ocean Facts," National Oceanic and Atmospheric Administration, December 2008, URL: http://oceanservice.noaa.gov/facts/oceandepth.html [cited 11 February 2010].

Vita

Major Anthony DeGregoria currently lives in Dayton, Ohio, with his wife and three children. They consider Florida their home, but move often, wherever the United States Air Force takes them. He received his Bachelor of Science degree in Mechanical Engineering with a Math Minor from the United States Air Force Academy in 2000.

Following his commission as an officer, Major DeGregoria served as an aircraft/store structural loads compatibility engineer. His other assignments included duties as an aircraft battle damage repair engineer, aircraft sustainment engineer, aircraft availability analyst, and aircraft maintenance officer.

Major DeGregoria received a Master of Science degree in Administration from Central Michigan University in 2008. Following his studies at the Air Force Institute of Technology in the Aeronautical Engineering program, he will work in the Air Vehicles Directorate of the Air Force Research Laboratory.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
25-03-2010	Master's Thesis	August 2008 – March 2010

4. TITLE AND SUBTITLE

Gravity Gradiometry and Map Matching:
An Aid to Aircraft Inertial Navigation Systems

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

DeGregoria, Anthony, Maj, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENY)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GAE/ENY/10-M06

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Sensors Directorate (AFRL/RYRN)
POC: Dr. Stewart DeVilbiss (stewart.devilbiss@wpafb.af.mil)
2241 Avionics Circle, Building 620
WPAFB OH 45433

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Inertial navigation systems (INS) offer passive, all-weather, and undeniable navigation information, which military customers often view as especially appealing strengths. Unfortunately, Airmen and engineers still struggle with INS's drifting position errors, and navigation aids generally detract from INS's strengths. At this year's Air, Space, and Cyberspace in the 21st Century Conference, the Chief of Staff of the Air Force identified the Global Positioning System (GPS) as a widely-known and exploitable vulnerability, saying that it's critical the Joint force reduce GPS dependence. Recent advances provide an opportunity for gravity gradient instruments (GGI), which measure spatial derivatives of the gravity vector, to aid an INS and preserve its strengths.

This thesis shows that a GGI and map matching enhanced (GAME) INS improves navigation accuracy, presents the conditions that make GAME feasible for aircraft, and identifies opportunities for improvement. The methodology includes computer models and algorithms, where a GGI and map matching aid an INS through a Kalman filter. Simulations cover different terrains, altitudes, velocities, flight durations, INS drifts, update rates, components of the gravity gradient tensor, GGI and map noise levels, map resolutions, and levels of interpolation. Although GAME with today's technology only appears worthwhile for long range and long endurance flights, the technologies expected in 10 years promise a broad spectrum of scenarios where GAME potentially provides great returns on investments and dominates the market for secure and covert navigation.

15. SUBJECT TERMS

navigation, aid, gravity, gravity gradiometry, INS, GPS, map matching

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Richard Huffman, Lt Col, USAF
U	**U**	**U**	**UU**	130	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636 x7490 (richard.huffmana@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18